



LAPIN YLIOPISTO  
UNIVERSITY OF LAPLAND

## **Ihan(a) kamala ohjelmointi**

Luova ohjelmointi lukion kuvataiteen opetuksessa

Kuvataidekasvatus  
Pro gradu -tutkielma

Jenna Hepoaho

18.5.2026  
Lapin yliopisto



LAPIN YLIOPISTO  
UNIVERSITY OF LAPLAND

Lapin yliopisto

Tiedekunta: Taiteiden tiedekunta

Työn nimi: Ihan(a) kamala ohjelmointi: Luova ohjelmointi lukion kuvataiteen opetuksessa

Tekijä: Jenna Hepoaho

Koulutusohjelma: Kuvataidekasvatus

Ohjaaja(t): TaT Annamari Manninen

Työn laji: Pro gradu -tutkielma

Sivumäärä, liitteiden lukumäärä: 61 + 3

Vuosi: 2026

## Tiivistelmä

Tässä pro gradu -tutkielmassa tarkastelen luovan ohjelmoinnin mahdollisuuksia osana lukion kuvataiteen opetusta. Tutkimuksen lähtökohtana on digitaalisen kulttuurin läpäisevyys nyky-yhteiskunnassa sekä tarve ymmärtää teknologiaa paitsi välineenä myös kriittisen ja taiteellisen ajattelun kohteena. Luova ohjelmointi nähdään tutkimuksessa ilmaisullisena, pedagogisena ja kulttuurisena ilmiönä, joka kytkeytyy post-digitaaliseen taidekasvatukseen.

Tutkimuksen tavoitteena on selvittää, miten luovan ohjelmoinnin opetuskokonaisuus voidaan toteuttaa osana lukion kuvataiteen opetusta. Tarkastelen erityisesti sitä, millaisia lukiolaisten kokemukset olivat luovan ohjelmoinnin opetuskokonaisuuden jälkeen.

Tutkimus toteutettiin taideperustaisena toimintatutkimuksena, jossa suunnittelin ja toteutin opetuskokeilun osana lukion nykytaiteen opintojaksoa. Tutkimusaineisto koostuu opetuskokonaisuuteen tuotetuista materiaaleista, opiskelijoiden p5.js-ohjelmointiympäristössä tuotetuista teoksista ja reflektioteksteistä sekä opetustilanteessa tehdyistä havainnoista. Analyysi painottuu opiskelijoiden reflektioteksteihin, joita tarkasteltiin laadullisen dialogisen tematisoinnin avulla. Tulosten perusteella luova ohjelmointi näyttäytyy monitasoisena ilmiönä, jossa ohjelmointi toimii samanaikaisesti teknisenä taitona, taiteellisen ilmaisun välineenä sekä digitaalisen kulttuurin kriittisen tarkastelun keinona. Lukio-opiskelijat hyödynsivät koodia visuaalisten ja symbolisten merkitysten rakentamisessa ja teoksissa käsiteltiin muun muassa algoritmeihin, sattumaan, kontrolliin ja ihmisen ja teknologian suhteeseen liittyviä teemoja. Ohjelmoinnin oppiminen koettiin sekä haastavana että palkitsevana ja siihen liittyi oivalluksia muun muassa minäpystyvyydestä, luovuudesta ja tekijyydestä.

Tutkimus osoittaa, että luova ohjelmointi voidaan integroida mielekkäästi osaksi kuvataiteen opetusta rajattuna ja kokemuksellisenä opetuskokonaisuutena. Post-digitaalisesta näkökulmasta tarkasteltuna se ei näyttäydy erillisenä teknisenä sisältönä vaan osana laajempaa tapaa ymmärtää, jäsentää ja luoda merkityksiä.

Avainsanat: luova ohjelmointi, kuvataidekasvatus, kuvataideopetus, ohjelmointi, post-digitaalisuus, taideperustainen toimintatutkimus, ohjelmoinnillinen ajattelu



LAPIN YLIOPISTO  
UNIVERSITY OF LAPLAND

University of Lapland  
Faculty: Faculty of Art and Design  
Title: Creative Coding in Upper Secondary School Art Education  
Author: Jenna Hepoaho  
Degree program: Art Education  
Thesis supervisor: DFA Annamari Manninen  
Type of work: Master's thesis  
Number of pages: 61 + 3  
Year: 2026

## Abstract

In this Master's thesis, I examine the potential of creative coding as part of upper secondary school art education. The study is grounded in the pervasive nature of digital culture in contemporary society and the need to understand technology not only as a tool but also as an object of critical and artistic inquiry. In this study, creative coding is understood as an expressive, pedagogical, and cultural phenomenon connected to post-digital art education.

The aim of the study is to investigate how a creative coding teaching sequence can be implemented as part of upper secondary school art education. In particular, I examine students' experiences following a creative coding learning module.

The study is based on a qualitative, art-based action research approach in which I designed and implemented a teaching intervention as part of a contemporary art course in upper secondary school. The research data consists of materials produced for the teaching module, artworks created by students in the p5.js programming environment, reflective texts, and observations made during the teaching sessions. The analysis focuses on students' reflective texts, which were examined using qualitative dialogical thematization.

Based on the findings, creative coding emerges as a multilayered phenomenon in which programming functions simultaneously as a technical skill, a medium for artistic expression, and a means of critically examining digital culture. Upper secondary school students used code to construct visual and symbolic meanings, and their works addressed themes related to algorithms, chance, control, and the relationship between humans and technology. Learning programming was experienced as both challenging and rewarding, and it gave insights related to self-efficacy, creativity, and authorship.

The study shows that creative coding can be meaningfully integrated into art education as a short and experiential learning sequence. From a post-digital perspective, it does not appear as a separate technical content area but as part of a broader way of understanding, structuring, and producing meanings.

Key words: creative coding, art education, programming, post-digital, art-based action research, computational thinking



## Sisällys

|          |  |           |
|----------|--|-----------|
| <b>1</b> | <b>Johdanto</b>  | <b>6</b>  |
| 1.1      | Taiteellinen ilmaisu taidekasvatuksessa ja nykytaiteessa                 | 9         |
| 1.2      | Ohjelmointi ja luova ohjelmointi taiteellisen ilmaisun lähtökohtana      | 10        |
| 1.3      | Post-digitaalinen aika ja digi-grasping                                  | 13        |
| 1.4      | Nykytaidepedagogiikka kietoutuneena ohjelmoinnin opettamisen menetelmiin | 15        |
| <b>2</b> | <b>Menetelmä ja analyysi</b>   | <b>16</b> |
| 2.1      | Taideperustainen toimintatutkimus  | 17        |
| 2.2      | Aineisto ja sen analyysi   | 19        |
| 2.3      | Tutkimuksen eettisyys  | 23        |
| <b>3</b> | <b>Luovan ohjelmoinnin opetuskokonaisuus kuvataiteen opetuksessa</b>     | <b>24</b> |
| 3.1      | Suunnittelun lähtökohdat ja tavoitteet                                   | 25        |
| 3.2      | Tuntien eteneminen   | 30        |
| <b>4</b> | <b>Opiskelijoiden kokemukset luovasta ohjelmoinnista</b>                 | <b>39</b> |
| 4.1      | Luova ohjelmointi taiteellisena ilmaisuna                                | 39        |
| 4.2      | Ohjelmointi oppimiskokemuksena   | 43        |
| 4.3      | Luova ohjelmointi merkityksellisenä kokemuksena                          | 48        |
| <b>5</b> | <b>Johtopäätökset</b>  | <b>50</b> |
| 5.1      | Ohjelmoinnin opettamisesta   | 50        |
| 5.2      | Miksi ohjelmointia?  | 52        |
| <b>6</b> | <b>Pohdinta</b>  | <b>54</b> |
| <b>7</b> | <b>Lähteet</b>   | <b>57</b> |
| <b>8</b> | <b>Liitteet</b>  | <b>61</b> |
| 8.1      | Liite 1. Padlet -kysymykset  | 61        |



|            |   |           |
|------------|---|-----------|
| <b>8.2</b> | <b>Liite 2. Opiskelijoiden reflektointitehtävän apukysymykset</b> | <b>61</b> |
| <b>8.3</b> | <b>Liite 3. Tutkimuslupa</b>                                      | <b>62</b> |

## 1 Johdanto

Tässä tutkimuksessa tuon luovan ohjelmoinnin osaksi lukion kuvataiteen opetusta. Luova ohjelmointi tarjoaa mahdollisuuden kurkistaa pintaa syvemmälle, sillä se tekee näkyväksi millaiseen ajatteluun ja logiikkaan laitteiden toiminta perustuu ja miten ”kone ajattelee”. Samalla luova ohjelmointi avaa taiteellisen näkökulman ohjelmointiin. Koodi ei ole vain tekninen väline vaan materiaali, jonka avulla voidaan ilmaista, kokeilla ja kysyä.

Taidekasvatuksen näkökulmasta tämä on merkittävää. Maailma muuttuu jatkuvasti, eikä taiteen opetus voi jäädä kehityksen ulkopuolelle. Luova ohjelmointi voi toimia siltana teknologian ja kriittisen ajattelun välillä, sillä sen avulla voidaan tarkastella digitaalista kulttuuria, valtarakenteita ja tulevaisuuden mahdollisuuksia taiteen keinoin. Luova koodaus ei tarjoa valmiita vastauksia, mutta se voi auttaa kysymään parempia kysymyksiä.

Olen itse elänyt digitaalisen kehityksen murroksessa ja nähnyt läheltä, miten koulumaailma on muuttunut digitalisaation myötä. Piirtoheittimet ja liitutaulut katosivat yllättävän nopeasti ja tilalle ilmestyi laitteita, joita kukaan ei tuntunut osaavan käyttää. Digitaalisten välineiden käyttöönotto oli monin paikoin haparoivaa, laitteet olivat läsnä, mutta niiden merkitys opetuksessa jäi usein epäselväksi.

Tänä päivänä tilanne on muuttunut. Digitaaliset laitteet ovat osa arkea ja koulua. Niiden käyttö sujuu pääosin vaivattomasti. Kommunikoimme laitteiden välityksellä, ohjaamme kotejamme sovelluksilla ja hyödynnämme tekoälyä yhä useammilla elämänalueilla. Robotit työskentelevät tehtaissa, siivoavat kotejamme ja kuljettavat ostoksia. Digitaalisuus ei ole enää poikkeus, vaan toimii jatkuvasti taustalla. Se on kietoutunut osaksi tätä elämää monella eri tasolla. Samalla herää kysymys: ymmärrämmekö, miten nämä laitteet ja järjestelmät toimivat? Miten ne kommunikoivat keskenään? Mihin niiden toiminta perustuu, miten ne on rakennettu ja millaisia arvoja ja oletuksia niihin on koodattu? Digitaalinen ympäristö tuntuu usein toimivan ikään kuin itsestään, näkymättömänä ja vaikeasti lähestyttävänä.

Oma kiinnostukseni aiheeseen kumpuaa henkilökohtaisista kokemuksistani. Kävin peruskoulun vuosina 2004–2013 eli ennen kuin ohjelmointi tuotiin osaksi perusopetuksen laaja-alaisia tavoit-

teita vuoden 2014 opetussuunnitelmassa. (Opetushallitus 2014). Toisin kuin myöhemmät ikäluokat, en ole saanut kosketusta ohjelmointiin koulupolullani vaan tutustuin siihen vasta aikuisiällä. Ensimmäiset askeleet ohjelmoinnin parissa otin vuonna 2021.

Alusta alkaen kiinnostukseni kohdistui koodin ilmaisullisiin mahdollisuuksiin. Kokeilin luoda kuvia, jotka muuttuivat tai paljastivat uusia kerroksia hiirtä liikuttaessa, sekä tekstejä, joita kutsuin koodirunoiksi (tätä tutkimusta tehdessäni törmäsin siihen, että se olikin ihan oikea termi eikä vain itseni keksimä). Näissä kokeiluissa koodi toimi sekä välineenä, että kielenä: runot olivat luettavaa kuten mikä tahansa teksti, mutta niiden merkitys avautui kokonaan vasta, kun tietokone suoritti kirjoitetun koodin. Vaikka nämä kokeilut olivat teknisesti yksinkertaisia, ne herättivät kysymyksiä näkyvästä ja piilotetusta.

Edellä mainitut kokemukset ovat ohjanneet minua pohtimaan luovan ohjelmoinnin paikkaa kuvataiteen opetuksessa, erityisesti lukion kontekstissa. Haluan tässä tutkimuksessa syventää ymmärrystäni luovasta ohjelmoinnista taiteellisena ja pedagogisena menetelmänä sekä tarkastella millaisia mahdollisuuksia se tarjoaa lukion kuvataiteen opetukselle. Tutkimuskysymykseni ovat:

1. Miten luovaa ohjelmointia voisi opettaa osana lukion kuvataiteen opetusta?
2. Miksi luovaa ohjelmointia kannattaa opettaa osana lukion kuvataiteen opetusta?

Tämän taiteellisen toimintatutkimuksen empiirinen aineisto rakentuu sekä tuottamistani oppimateriaaleista ja omista muistiinpanoistani sekä opiskelijoiden teoksista ja reflektiosta. Pääaineisto on opiskelijoiden reflektiotekstit, koska niiden avulla saan paitsi tietoa, miten luovan ohjelmoinnin kokonaisuus onnistui, mutta myös arvokasta tietoa mitä ajatuksia lukio-opiskelijoille rakentuu luovan ohjelmoinnin, oman oppimisen ja oman taiteellisen prosessin ympärille. Näin opiskelijoiden kokemukset luovan ohjelmoinnin tuntikokonaisuudesta eivät ainoastaan kerro miten opetusta kannattaisi toteuttaa, vaan myös miksi sitä kannattaa toteuttaa. Nämä kaksi tutkimuskysymystä tukevat toinen toisiaan. Pelkästään toista tarkastelemalla tutkimus jäisi latteaksi. Molempia kysymyksiä tarkastelemalla saadaan kokonaiskuva tutkimuksen aiheesta. Asetankin tämän tutkimuksen hypoteesiksi sen, että luovan ohjelmoinnin kokonaisuus voisi paitsi kehittää opiskelijoiden kykyä ymmärtää koodia myös herätellä ajattelemaan kriittisesti teknologiaa ja digitaalista ympäristöä, kun opetus rakentuu tukemaan paitsi koodinlukutaitoa, mutta myös liittämään aiheet lukio-opiskelijan omaan elämismailmaan.

Knochel ja Patton (2015) ovat avanneet keskustelun digitaalisuuden ymmärtämisen tärkeydestä osana taidekasvatusta jo reilu kymmenen vuotta sitten. Heidän ajatuksensa ovat edelleen ajan-

kohtaisia. He nostivat artikkelissaan hyvin esiin sen, kuinka digitaalisuus jää ymmärtämättä käytännön toiminnan tasolla. Esimerkkinä nostettiin muun muassa se kuinka taideopiskelijat eivät osaa selittää tai kuvailla kuinka jokin videopeli tai mobiiliappi toimii, kun taas keramiikan poltto-prosessin selittäminen luonnistuu tai keskustelu eri pigmenttien vaikutuksesta sävyyn ja valööriin. (Knochel & Patton 2015, 22.) Toisaalta pitää ottaa huomioon, että he puhuvat nimenomaan taideopiskelijoista eikä yleisesti oppijoista, joten syvälinen ymmärtäminen väriopista tai keramiikasta ei ole oletus jokaisen peruskoululaisen tai lukiolaisen kohdalla. Lähinnä ehkä haen tässä sitä, että kuvataiteen opetuksessa käydään läpi jollain tasolla niin värioppia kuin keramiikkaa ja näiden tekniikoita. Kun taas digitaalisia välineitä hyödynnettäessä voidaan ajatella, että käytännön toiminnan tasolla ymmärtäminen riittää eli miten jokin asia toimii sen sijaan, että miksi se toimii tietyllä tapaa ja mitä siihen toimintaan liittyy. Esimerkiksi kuvankäsittelyä opettaessa huomio voi kiinnittyä siihen, kuinka oppija oppii säätämään värejä haluamansa lopputuloksen saavuttamiseksi. Tällöin huomio on ohjelman käyttöliittymässä eikä siinä minkälaisen algoritmien ja oletusten varaan värien säädöt rakentuvat ohjelmassa. Kriittisen tarkastelun kohteena voisi sen sijaan olla miten automaattiset värinkorjaukset tai suodattimet tuottavat tietynlaisia visuaalisia ihanteita ja rajaavat näin ollen mahdollisia ilmaisullisia ratkaisuja. Tällä tavoin voidaan oppia kuvankäsittelyä, mutta myös havahtua tarkastelemaan sitä kriittisesti. Syy siihen, miksi tämäntyyppinen lähestymistapa on jäänyt varjoon kuvataidekasvatuksen kentällä voi johtua siitä, että tutkimusta tai opetusmateriaaleja ei ole tarpeeksi tarjolla ja digitaaliset välineet ja teknologia ylipäätään koetaan vieraana ja vaikeana.

“—harvat taideopettajat ovat osoittaneet osaamista tietokoneohjelmoinnissa, joten alalla on suurilta osin vältetty ohjelmoitavien objektien toiminnan kriittistä tarkastelua ja sellaisten teknisten valmiuksien kuin tietokoneohjelmoinnin kehittämistä taiteellisenä työskentelytapana” (tekijän suomentama; Knochel & Patton 2015, 22)

Tämä tutkimus asettuu osaksi tätä keskustelua ja pyrkii osaltaan avaamaan tilaa luovalle ohjelmoinnille taiteellisen ilmaisun ja ajattelun välineenä sekä materiaalina. Toivon, että tutkimuksena madaltaa kynnystä käsitellä luovaa ohjelmointia osana kuvataiteen opetusta.

Aikaisemmin kuvatut kokemukset ja kysymykset asettuvat laajempaan keskusteluun taidekasvatuksen, digitalisaation ja teknologisen ymmärtämisen suhteesta. Jotta luovaa ohjelmointia voidaan tarkastella kuvataiteen opetuksen kontekstissa mielekkäällä tavalla, on tarpeen jäsentää niitä teoreettisia näkökulmia, joiden varaan tutkimus rakentuu. Luova ohjelmointi ei ole yksiselitteinen ilmiö, vaan se sijoittuu useiden käsitteellisten kenttien risteykseen: taiteelliseen ilmaisuun, ohjelmointiin, pedagogiikkaan sekä digitaalisen kulttuurin kriittiseen tarkasteluun. Näin voidaan pyrkiä ymmärtämään, millaisena ilmaisun muotona luova ohjelmointi voidaan nähdä,

millaisia pedagogisia mahdollisuuksia siihen liittyy ja millaista suhdetta se rakentaa digitaaliseen maailmaan.

### **1.1 Taiteellinen ilmaisu taidekasvatuksessa ja nykytaiteessa**

Taiteellinen ilmaisu on keskeinen osa taidekasvatusta, jossa korostuu samanaikaisesti yksilön oman ilmaisun tukeminen sekä taiteen laajempi kulttuurillinen ja yhteiskunnallinen merkitys. Nykytaidetta ei määritä yhtenäinen tyyli tai tekniikka vaan pikemminkin monimuotoisuus, prosessinomaisuus ja merkityksellisyys. Teoksen arvo ei tällöin perustu ensisijaisesti muotoon tai tekniseen taituruuteen vaan siihen, mitä teos tekee tai kysyy ja millaisia suhteita se synnyttää tekijän, katsojan ja ympäröivän maailman välille. (Tieteen termipankki 2026; Kiasma 2026.) Tästä syystä nykytaide muodostaa taiteelliselle ilmaisulle erityisen kontekstin.

Taiteellisella ilmaisulla tarkoitan siis tässä yhteydessä sitä, että taiteellisen tekemisen kautta ilmaistaan esimerkiksi omaa kulttuuria, ajatuksia ja tunteita yksilön oman kokemuksen ja tulkinnan kautta. Taiteellinen tekeminen kattaa laajasti erilaisia ilmaisun muotoja kuvallisesta performatiiviseen ilmaisuun. Lukion opetussuunnitelmassa ilmaisulla on keskeinen asema ja lukio-opetuksen yhteisenä tavoitteena on vahvistaa lukio-opiskelijan ilmaisutaitoja (Opetushallitus 2019, 38). Ilmaisua ymmärretään laajasti kielellisenä, kuvallisena ja kehollisena toimintana. Kiinnostavaa kuitenkin on, että opetussuunnitelmassa kuvataiteen osalta puhutaan kuvien tuottamisesta, kuvailmaisusta ja ilmaisukeinoista, kun taas musiikin kohdalla korostetaan taiteellista ilmaisua (Opetushallitus 2019, 157–163; 178–180).

Taiteellinen ilmaisu on prosessi, joka rakentuu tekemisen prosessin ja teoksen välisessä suhteessa. Kun yksilö ilmaisee itseään taiteellisesti se ei ole aina ulkopuoliselle suoraan koettavissa. Esimerkiksi kuvataiteen luokassa opettaja voi olla tästä ilmaisusta etäällä, jolloin sen kokeminen jää väistämättä rajalliseksi, koska päästään oikeasti lähelle vain lopputulosta, kun taas itse prosessi voi jäädä etäiseksi kokemukseksi. Valmiista teoksesta voidaan pyrkiä tulkitsemaan, mitä tekijä on pyrkinyt ilmaisemaan teoksellaan. Lukio-opiskelijan mielensisäistä prosessia voidaan pyrkiä ymmärtämään sosiaalisen vuorovaikutuksen avulla, reflektiotekstillä ja näiden kautta arvioida itse ilmaisua. Tässä taide ja matematiikan opiskelu luo yllättävän yhtenäisyyden – ilman vaatimusta välivaiheiden esittämistä matematiikan tehtävissä emme pysty ymmärtämään ajatusprosessia. Samoin kuin kuvataiteessa ilman ymmärrystä sisäisestä ajatuksesta prosessin keskellä emme pysty havainnoimaan taiteellisen ilmaisua. Puhuttaessa taiteellisesta ilmaisusta en siis tarkoita pelkästään aistein koettavaa lopputulosta vaan kokemuksellista, kehollista, sosiaalista ja reflektioivaa prosessia, jossa oppija rakentaa suhdetta itseensä, toisiin ja maailmaan.

Tällainen ilmaisu avaa tilaa myös uusille materiaaleille ja välineille, kuten digitaaliseen teknologialle ja sitä kautta ohjelmoinnille.

Ajattelen siis, että taideteos voidaan ymmärtää taiteellisen toiminnan materialisoituneena tai dokumentoituneena ilmentymänä ja taiteellinen ilmaisu itse prosessina, jota myös itse teos on osana. Vaikka taiteellinen ilmaisu on ajallista, kehollista ja usein vain osittain ulkopuolisen koettavissa, se jäsenyy osaksi itse teosta. Nämä ajatukseni mukailevat John Deweyn (2010, 95–96) ajatusta siitä, että taiteellinen ilmaisu ei ole vain lopputulos vaan prosessi, jossa kokemus muotoutuu tekemisen ja materiaalisten ehtojen vuorovaikutuksessa yhtenäiseksi kokonaisuudeksi. Taiteelliseen ilmaisuun kytkeytyy syvästi kokemus. Yksilön kokemus ympäröivästä todellisuudesta, mielikuvitelmista ja itsestään – eletystä elämästä. Deweyn (2010) mukaan yksilön kokemukseen ilmaisu kytkeytyy siten, että nykyinen tekeminen on jatkuvassa yhteydessä niihin merkityksiin ja arvottamiseen tapoihin, joita eletty elämä on muovannut. Ilmaisu ei ole oikeastaan kokemusten esittämistä vaan niiden jäsentymistä ja muuntumista prosessissa. Se edellyttää rajoitteita sillä se ei ole vapaata purkautumista vaan niiden rajoitteiden puitteissa tapahtuvaa muotoutumista. (Dewey 2010, 86–91.)

Deweyn (2010) kokemuksellinen ja ilmaisullinen ajattelu korostuu Marjo Räsäsen (2008) kehittämässä kokemuksellis-konstruktivistisen taideoppimisen mallissa. Mallissa taideoppiminen ymmärretään prosessinsa, jossa oppija rakentaa ymmärrystään aktiivisesti omien havaintojensa, kokemustensa ja reflektion kautta. Oppiminen ei perustu valmiin tiedon vastaanottamiseen vaan kokemusten tulkintaan ja niiden merkitysten jäsentämiseen. Räsäsen (2008, 106–107) mukaan reflektio onkin taiteellisen oppimisen perusta: oppija tarkastelee kokemuksiaan ja niiden merkityksiä, joiden ilmentymänä myös taideteos voidaan nähdä. Räsäsen mallissa taideoppiminen ei tähtää ainoastaan erilaisten taitojen omaksumiseen vaan enemmin laajempien tietorakenteiden muodostumiseen. Tämän vuoksi Räsäsen (2008, 107–108) mukaan kuvataiteen opetus tulisi kytkeä muihin tiedonaloihin sekä yhteiskunnan käytäntöihin. Näin taidekasvatus ei näyttäyty irrallisena oppiaineena vaan tapana ymmärtää maailmaa suhteessa kulttuuriin ja yhteiskuntaan. Räsäsen (2008) mukaan taideoppiminen voi muuttaa oppijan suhdetta muun muassa tarkasteltavaan ilmiöön ja kulttuuriin.

## **1.2 Ohjelmointi ja luova ohjelmointi taiteellisen ilmaisun lähtökohtana**

Fagerlund (2022, 125) toteaa väitöskirjaansa nojaten, että ”jokaisella oppilaalla tulisi olla oikeus oppia ohjelmointia ja ohjelmoinnillista ajattelua aina esiopetuksesta korkeakoulutukseen”. Luokion opetussuunnitelmassa ohjelmointi on valinnaisena aineena (Opetushallitus 2019). Yleisesti

ohjelmointi lukiossa on tarjolla edellisen opetussuunnitelman aikaan tehdyssä selvityksessä yleisimmin tietotekniikassa (riippuen lukiosta onko erillisenä oppiaineena) ja pitkässä matemaatiikassa (Nousiainen & Kivistö 2022). Tämän voidaan ajatella olevan hieman ristiriitaista, koska ohjelmointi on osa perusopetuksen laaja-alaisia tavoitteita (Opetushallitus 2014). Miksei tätä jatkumoa ole viety lukio-opetukseen?

Ohjelmoinnin ajatellaan usein olevan tekninen taito, jossa kirjoitetaan koodia matemaattisten algoritmien pohjalta. Tällainen näkökulma kuitenkin kaventaa ohjelmoinnin merkitystä. Knochel ja Patton (2015) esittävät, että ohjelmointi on aina sosiaalisten, poliittisten ja kulttuuristen puitteiden kautta tapahtuva suunnitteluprosessi. Koodi ei synny tyhjiössä vaan sen heijastaa arvoja, valintoja ja valtasuhteita. Tämä näkökulma on keskeinen taidekasvatuksen ja luovan ohjelmoinnin kannalta. Kun koodi nähdään kulttuurisena materiaalina, siitä tulee mahdollinen ilmaisun ja kriittisen tarkastelun väline. Ohjelmointi ei tällöin rajaudu ongelmanratkaisuun vaan avaa tilaa merkityksenannolle, kokeilulle ja rajojen siirtämiselle.

Ohjelmoinnin tarkoitus opetuksen näkökulmasta katsottuna ei ole pelkästään se, että opiskelija oppisi ohjelmoimaan vaan se, että opiskelija havahtuisi kriittisesti tarkastelemaan siihen liittyviä asioita (Mertala ym. 2020). Kriittinen tarkastelu voi synnyttää transformatiivista oppimista. Transformatiivinen oppiminen tarkoittaa lyhyesti ajattelutapojen muutosta. Tässä ajattelun prosessissa yksilö ei ainoastaan opi uutta vaan muuttaa sitä ”linssiä”, jonka läpi hän ylipäättään tulkitsee maailmaa. Transformatiivisen oppimisen keskiössä on kriittinen itsereflektio, josta syntyy kokemuksen uudelleen tulkinta, joka johtaa toiminnan muutokseen. Eli ytimessä on koetella ja muokata yksilön syvälle juurtuneita ajattelu- ja toimintamalleja. (Anttila 2022, 1:3.)

Ohjelmointiin liitetään myös vahvasti ohjelmoinnillinen ajattelu. Ohjelmoinnillinen ajattelu (computational thinking) viittaa ajattelutapaan, jossa monimutkaisia ilmiöitä jäsennetään abstrahoinnin, algoritmien ja simulaation avulla (Wing 2006). Suomessa ohjelmoinnillinen ajattelu on integroitu osaksi perusopetuksen laaja-alaista osaamista, eikä sitä opeteta tällöin erillisenä oppiaineena vaan osana eri oppiaineita (Opetushallitus 2014). Opetus- ja kulttuuriministeriö on julkaissut vuonna 2022 digitaalisen osaamisen kuvaukset, joihin myös ohjelmoinnillinen ajattelu sisältyy. Julkaisu on osa Uudet lukutaidot -kehittämishjelmaa, jonka tavoitteena on medialukutaidon, tieto- ja viestintäteknologian ja ohjelmoinnin osaamisen vahvistaminen varhaiskasvatuksesta esi- ja perusopetukseen (Opetushallitus 2022; Mediataitokoulu 2026). Näiden tavoitteiden pohjalta ohjelmoinnillinen ajattelu voidaan huomioida selkeämmin osana opetusta.

Luovalle ohjelmoinnille on kehitetty omia ohjelmointiympäristöjä. Nämä visuaalisesti ajatteleville tekijöille suunnatut ohjelmointiympäristöt, kuten Logo, Max, Scratch ja Processing ovat historiallisesti pyrkineet madaltamaan kynnystä ohjelmointiin ja laajentamaan käsitystä siitä, kelle ohjelmointi kuuluu (Reas & Fry 2007, 1–2). Näissä ympäristöissä koodi ei ole vain tekstiriviä vaan visuaalinen, auditiivinen ja vuorovaikutteinen materiaali. Processing on ollut pitkään melko suosittu ohjelmointiympäristö. Tästä kertoo ehkä se, että muun muassa opetushallituksen sivuilta löytyy valmiita tehtäviä tehtäväksi Processingilla. Nykyään luovaan ohjelmointiin on tarjolla monenlaisia alustoja, eikä koodia tarvitse välttämättä kirjoittaa itse vaan voidaan raahata tai yhdistellä erilaisia palikoita. Esimerkiksi aikaisemmin mainitussa Scratch -ympäristössä on valmiita laatikoita, joita yhdistelemällä rakennetaan koodia. Scratchin lisäksi on olemassa hieman samantyyppinen, mutta monimutkaisempi ohjelmointiympäristö nimeltään Cables.

Luova ohjelmointi tarkoittaa ohjelmoinnin hyödyntämistä taiteellisen ja ilmaisullisen työskenteilyn välineenä eikä siinä ole tarkoitus keskittyä niinkään ohjelman toiminnallisuuteen tai teknisiin yksityiskohtiin (Terroso & Pinto 2022, 13:2). Siinä koodi rinnastuu muihin taiteen materiaaleihin, kuten esimerkiksi siveltimen vetoon, sanan rytmiin tai musiikilliseen fraasiin. Luovan ohjelmoinnin historia juontuu aina 1960-luvun digitaalisen taiteen kokeiluihin (Chibalashvili ym. 2023), mutta sen merkitys on kasvanut erityisesti post-digitaalisessa kontekstissa. Luovan ohjelmoinnin on todettu olevan tehokas tapa oppia ohjelmointia (Greenberg ym. 2012; Woodym. 2016). Samalla luova ohjelmointi toimii myös luonnollisena tapana oppia koodin lukutaitoa. Dufva & Dufva (2016, 98) mainitsevat, että koodin lukutaito (code literacy) ei viittaa suoraan ohjelmoinnin oppimiseen perinteisessä mielessä, vaan se tarkoittaa koodin sekä sen tarkoitusten ja kontekstin ymmärtämistä. Koodin ymmärtäminen ei synny luonnostaan elämänkokemuksen kautta, vaan se on tietoisesti opetettava. (Dufva & Dufva 2016, 98).

Dufva (2018) kuvaa luovaa ohjelmointia sekä taiteellisenä käytäntönä että menetelmänä ymmärtää digitaalista maailmaa. Koodaamalla ei ainoastaan tuoteta teoksia, vaan rakennetaan suhdetta teknologiaan. Tässä suhteessa koodi ei ole neutraali materiaali vaan se on osa kulttuurisia, taloudellisia ja poliittisia rakenteita (Slotte Dufva 2021). Jotta koodin ulottuvuutta ja monimuotoisuutta voisi ymmärtää paremmin Dufva & Dufva (2016) ovat jakaneet koodin yhdeksään eri metaforaan, jotka pitävät sisällään eri näkökulmia ohjelmointiin liittyen. Tämän tutkimuksen kannalta oleellisimpina metaforia koodille ovat karnevaali ja kulttuuri.

Karnevaali korostaa koodin kykyä olla taiteen ja luovuuden väline, jonka kautta haastetaan ajattelutapoja ja luodaan keskustelua. (Dufva 2018, 34). Eli juuri sitä, mitä taiteellisen ilmaisun kautta voidaan tavoitella. Knochel ja Patton (2015) korostavat, että taidekasvatuksessa koodin käyttö

voi avata tilaa teoksille, jotka ovat vuorovaikutuksessa ympäristönsä kanssa. Kulttuuri taas ymmärtää koodin ajattelun ja ymmärtämisen välineenä, jonka tarkoitus on luoda yhteyksiä ja rakentaa yhteisöjä. Tämä vertaus keskittyy erityisesti arvoihin, uskomuksiin ja maailmankuvioihin. (Dufva & Dufva 2016, 102–103.) Esimerkiksi avoimen lähdekoodin yhteisöissä, kuten Linux-kehityksessä tai Wikipedia-projekteissa, koodi ei näytkään pelkästään teknisenä ohjeistuksena vaan yhteisesti jaettuna tapana tuottaa ja arvioida tietoa. Näissä yhteisöissä keskeiset arvot kuten avoimuus, yhteistyö ja vertaisarviointi ohjaavat sekä koodin kirjoittamista että sen tulkintaa. Tällöin koodi toimii kulttuurisena käytäntönä, joka rakentaa yhteisön toimintalogiikkaa ja määrittää millaista tietoa pidetään oikeutettuna. Toinen esimerkki löytyy sosiaalisen median algoritmeista. Niiden ”koodi” ei ainoastaan järjestä sisältöä vaan myös heijastaa kulttuurisia oletuksia siitä, mikä on kiinnostavaa, arvokasta tai sitouttavaa. Esimerkiksi se, että tietyn tyyppinen sisältö priorisoidaan näkyvyyden kustannuksella rakentaa samalla käsityksiä huomiosta, menestyksestä tai jopa sosiaalisesta arvosta.

### **1.3 Post-digitaalinen aika ja digi-grasping**

Digi-graspingin käsite tuo tarkasteluun kokemuksellisen ja kehollisen ulottuvuuden, jossa digitaalinen ymmärrys rakentuu tekemisen, kokeilun ja vuorovaikutuksen kautta (Dufva & Dufva 2019). Post-digitaalinen näkökulma puolestaan sijoittaa luovan ohjelmoinnin ajankohtaiseen kulttuuriseen ja yhteiskunnalliseen kontekstiin, jossa digitaalinen teknologia on kaikkialla läsnä, mutta samalla kriittisen tarkastelun kohteena (Slotte Dufva 2021).

Post-digitaalisuuden käsitettä voidaan tarkastella tilanteena, jossa digitaalisen ja ei-digitaalisen välinen raja ei ole enää selkeä, vaan ne kietoutuvat toisiinsa monin tavoin. Saethre-McGuirkin (2022) mukaan post-digitaalinen ei viittaa digitaalisen teknologian jälkeiseen aikaan vaan pikemminkin tilaan, jossa digitaalisuus on sulautunut osaksi arkea, vuorovaikutusta ja taiteellista toimintaa. Myös Slotte Dufva (2021, 270–271) näkee post-digitaalisen aikakauden samankaltaisena, kriittisesti tarkasteltavana ilmiönä, jossa digitaalinen teknologia nähdään ulottuvan kaikille. Digitaalinen ei ole erillinen kerros vaan kietoutunut materiaaliin, ekologisiin ja yhteiskunnallisiin kysymyksiin. Post-digitaalista ei pidä sekoittaa jonkin ajanjakson päättymiseen. Cramer (2014, 13) huomauttaa, että tässä yhteydessä ”post” tarkoittaa jatkumoa ja muutosta. Digitaalinen ei siis ole kadonnut vaan jatkuu osana arkea samalla kun sen merkitykset ja käyttötavat muuttuvat ja tulevat kriittisen tarkastelun kohteeksi. Samalla post-digitaalinen voidaan ymmärtää myös muutoksena asenteissa teknologiaa kohtaan. Se viittaa tilanteeseen, jossa aiempi teknologinen innostus on alkanut muuttua kriittisemmäksi ja paikoin myös jopa kyllästyneeksi suhtautumiseksi digitaalisiin järjestelmiin ja laitteisiin (Cramer 2014, 12).

Taidekasvatuksen kontekstissa post-digitaalisuus merkitsee sitä, ettei digitaalista työskentelyä tule käsitellä erillisenä osa-alueena vaan osana kokonaisvaltaista oppimista ja tekemistä. Saethre-McGuirk (2022, 74) painottaa, että nykyaikaisen ja oppijoiden arjen näkökulmasta digitaalinen ja aineellinen toiminta kietoutuvat yhteen, jolloin opetuksessa on tärkeää huomioida niiden välinen jatkuva vuorovaikutus. Post-digitaalinen taidekasvatus tarkoittaa siis digitaalisuuden ymmärtämistä osana laajempaa tapaa olla, toimia ja luoda merkityksiä.

Luovan ohjelmoinnin opetuskokonaisuus on osa post-digitaalista taidekasvatusta, jossa digitaalinen teknologia ei näy erillisenä sisältöalueena, vaan kietoutuu osaksi laajempaa tapaa tehdä, kokoa ja ymmärtää maailmaa. Tämä ilmenee siinä, että ohjelmointi toimii samanaikaisesti taiteellisen ilmaisun välineenä, kokemuksellisenä työskentelyprosessina sekä digitaalista kulttuuria koskevan ajattelun ja reflektion kohteena. Lukio-opiskelijat eivät ainoastaan opettele teknisiä taitoja vaan rakentavat tekemisen kautta suhdetta digitaalisiin ilmiöihin, kuten algoritmeihin, interaktiivisuuteen ja teknologian rooliin arjessa. Tällöin digitaalinen asettuu osaksi oppijan maailmasuhdetta. Se ei toimi irrallisena työkaluna vaan se on tapa toimia, hahmottaa ja rakentaa merkityksiä ympäröivälle todellisuudelle. Luovan ohjelmoinnin avulla taiteellinen ilmaisu voi tehdä näkyväksi teknologian piilossa olevia rakenteita. Esimerkiksi laitteiden purkaminen ja niiden osista rakentaminen taiteelliseksi kokonaisuuksiksi paljastaa digitaalisen kulttuurin materiaalisesta perustasta ja ekologiset seuraukset (Slotte Dufva 2021, 277–279). Taiteellinen ilmaisu ei tällöin jaa pintatasolle, vaan toimii kriittisenä taiteellisenä tutkimuksena.

Digi-grasping tarkoittaa tapaa ymmärtää digitaalisuutta kokemuksellisesti ja kehollisesti ilman, että ymmärrys perustuu ensisijaisesti käsitteisiin, teorioihin tai tekniseen tietoon. Kyse ei siis ole siitä, että ihminen osaisi selittää, miten digitaaliset järjestelmät toimivat vaan siitä, että hän aistii, tunnistaa ja hahmottaa digitaalisuuden osana omaa elämäänsä ja ympäristöään. (Dufva & Dufva 2019.)

Määritelmämme mukaan digi-graspingilla on erilaisia ilmenemismuotoja, mutta yleisesti ottaen digi-grasping voidaan määritellä aktiiviseksi ja voimaannuttavaksi merkityksenmuodostukseksi ja osallistumiseksi yhä digitalisoituvassa maailmassa, joka ei perustu pelkästään rationaaliseen ymmärrykseen, vaan myös keholliseen ymmärrykseen. On syytä huomata, että digi-grasping ei pyri määrittelemään olemista virtuaalisessa maailmassa, vaan se on kiinnostunut enemmän fyysisestä maailmasta, joka digitalisoituu yhä enemmän. (tekijän suomentama; Dufva & Dufva 2019, 20)

Digi-graspingin avulla ihmiset voivat ymmärtää ja kyseenalaistaa nykyisten digitaalisten rakenteiden taustalla olevia valintoja ja motiiveja sekä luoda uusia rakenteita. Nykyajan palveluiden ja elämisen ollessa yhä enemmän teknologian varassa, digi-graspingia voidaan pitää tärkeänä lähestymistapana digitaalisen yhteiskunnan tulevaisuuden muovaamisessa (Dufva & Dufva 2019 17). Tämä

näkökulman vuoksi digi-graspingista voisi olla hyötyä koulutuksessa. Sen kautta voidaan laajentaa ymmärrystä digitaalisuudesta pelkän teknisen osaamisen ulkopuolelle. Digi-graspingin olemisen ja toimimisen tapojen kautta voi olla helpompi ymmärtää digitalisaation laajuutta (Dufva & Dufva 2019, 26). Näin digi-graspingilla on vahva yhteys post-digitaalisuuteen, sillä se painottaa ymmärrystä ja tiedostamista arjessa teknisen osaamisen sijaan. Ymmärtämisellä tarkoitetaan sitä, että yksilö on tietoinen digitaaliset järjestelmien ja teknologian roolista ja vaikutuksista jokapäiväisessä elämässä (Dufva & Dufva 2019, 19). Dufva & Dufva (2019, 19) huomauttavat, että ymmärtämiseen liittyy myös kehollinen ulottuvuus, jossa digitaalinen maailma tulee ”tartutuksi” (grasp) toiminnan, kokemisen ja tekemisen kautta eikä ainoastaan käsitteellisen ajattelun tasolla. Tämä korostuu, koska digitaalinen on usein käyttäjälle näkymätöntä. Tällöin järjestelmien toimintaa ja niiden vaikutusta omaan toimintaan ei välttämättä tiedosteta. (Dufva & Dufva 2019, 19).

#### **1.4 Nykyaidepedagogiikka kietoutuneena ohjelmoinnin opettamisen menetelmiin**

Opetuksen keskeisenä pedagogisena ratkaisuna tässä tutkimuksessa käytettiin reaaliaikaista koodauksen (eng. live coding) menetelmää, jossa opettaja kirjoittaa koodia reaaliaikaisesti opiskelijoiden nähden ja selittää samalla sen toimintalogiikkaa. Menetelmän pedagoginen arvo perustuu siihen, että ohjelmoinnin ajatteluprosessi tehdään näkyväksi ja oppimisen keskeneräisyys sallitaan (Brown & Wilson 2018, 3).

Reaaliaikainen koodaus ei ole yksisuuntainen demonstraatio vaan vuorovaikutteinen ja prosessikeskeinen työskentelytapa, joka mahdollistaa yhteisöllisen tiedon rakentamisen. Reaaliaikainen koodaus on yksi Fuentes Martinezin (2024) väitöskirjassaan nostamista yleisesti käytössä olevista opetusmetodeista ohjelmoinnin saralla. Reaaliaikaisen koodauksen puolesta puhuu myös Brown ja Wilson (2018). He nostavat esille muun muassa sen, että se pakottaa opettajan hidastamaan tahtia verrattaen diaesityspohjaiseen opetukseen (Brown & Wilson 2018, 3). Reaaliaikaisen koodauksen lisäksi käytössä oli myös Fuentes Martinezin (2024, 23–25) toinen esille nostama metodi, nikkarointi (tinkering), jossa oppilaiden on tarkoitus tutkia ja muokata heille annettua ohjelmakoodia.

Tämä lähestymistapa kytkeytyy nykyaidepedagogiikan periaatteisiin, jossa oppiminen nähdään kokemuksellisena, prosessuaalisena ja sosiaalisena toimintana. Nykyaidepedagogiikka korostaa kokeilun sallimista, virheiden merkitystä oppimisessa ja moninaisuuden arvostamista. (Saastamoinen 2011; Juntunen 2011; Anttila 2022, luku 7.) Se sijoittaa oppijan merkityksen rakentajana

ja toimijana prosessiin. Reaaliaikainen koodaus mahdollistaa tämän pedagogisen lähestymistavan käytännössä. Lukio-opiskelijat seuraavat koodin syntymistä reaaliajassa, kirjoittavat sitä samanaikaisesti omille koneilleen ja voivat heti lähteä kokeilemaan ja leikkimään koodilla. Opettajaohjoinen vaihe toimii pedagogisena tukena, jonka avulla lukio-opiskelijat voivat myöhemmin soveltaa ja kehittää omaa työtään itsenäisesti.

## 2 Menetelmä ja analyysi

Tämä tutkimus toteutettiin laadullisena taideperustaisena toimintatutkimuksena, jossa tutkimuksen kohteena oli luovan ohjelmoinnin opetuskokonaisuuden suunnittelu, toteutus ja arviointi osana lukion kuvataiteen opetusta. Tutkimus toteutettiin syksyllä 2025 eteläsuomalaisessa lukiossa osana nykytaiteen opintojaksoa. Luovan ohjelmoinnin opetuskokonaisuus sijoitettiin osaksi olemassa olevaa opintojaksoa, joten tutkimus toteutettiin autenttisisessa lukiokontekstissa osana normaalia opetusta. Opetuskokonaisuus koostui yhteensä viidestä 75 minuutin opitunnista, joista kolme ensimmäistä oli varattu luovan ohjelmoinnin opettamiseen ja viimeiset kaksi lukio-opiskelijoiden kiireettömään työskentelyyn tuntikokonaisuuden Koodattu maailma -tehtävän parissa. Työskentelyssä käytettiin OpenProcessing-ohjelmointiympäristöä ja itse ohjelmointi toteutettiin p5.js-kirjaston avulla JavaScript-kielellä.

Laadullisessa tutkimuksessa keskeisenä empiirisenä aineistona toimii esimerkiksi erilaiset tekstit ja kuvat (Juhila 2021). Tutkimukseni aineisto on rajattu koskemaan lukio-opiskelijoiden reflektointikirjoitelmia. Tuntikokonaisuuden aikana tuotetut teokset toimivat näiden ajatusten tutkimateriaalina. Kirjoitettujen tekstien avulla on mahdollista tarkastella opiskelijoiden ajatteluprosesseja, kokemuksia ja oppimista tuntikokonaisuuden aikana, kun taas teokset toimivat tukevinä visuaalisina havaintoina. Reflektiot toimivat siten luovan ohjelmoinnin opetuskokeilun palautteena sekä tutkimusaineistona. Tutkimuskysymys asettui laadullisen tutkimuksen mukaisesti, jossa miten ja miksi kysymykset painottuvat usein (esim. Juhila 2021). Tutkimuskysymykseni ovat: miten luovaa ohjelmointia voisi opettaa osana lukion kuvataiteen opetusta ja miksi luovaa ohjelmointia kannattaa opettaa osana lukion kuvataiteen opetusta. Tutkimukseni mukailee Jokela & Huhmarniemen (2020, 45) esittelemää taideperustaisen toimintatutkimuksen toteuttamisen vaiheistusta, joka koostuu alkukartoituksesta ja suunnittelusta, taiteellisesta toiminnasta, aineiston rakentumisesta toiminnan yhteydessä tai sen jälkeen ja aineiston analysoinnista.

## 2.1 Taideperustainen toimintatutkimus

Pelkkä toimintatutkimus ei kuvaa tutkimukseni luonnetta riittävän tarkasti. Taideperustaisen toimintatutkimuksen juuret ovat sekä toimintatutkimuksessa että taiteellisessa tutkimuksessa (Jokela & Huhmarniemi 2020, 40). Molempien periaatteisiin kuuluu uuden toimintatavan kehittäminen, mutta taideperustaisen toimintatutkimuksen aineisto rakentuu taiteellisen toiminnan kautta, mikä erottaa taideperustaisen toimintatutkimuksen olennaisesti muista samankaltaisista menetelmistä (Hakala 2024, 93; Jokela & Huhmarniemi 2025, 109). Tämä lähtökohta sopi tutkimukseeni sen suunnitteluvaiheessa, kun tavoitteeni oli kehittää ja kokeilla mielekästä tapaa integroida luova ohjelmointi osaksi lukion kuvataiteen opetusta. Luovan ohjelmoinnin opetuskokonaisuuden suunnittelu ja toteutus sijoittuu opettamisen ja taiteen piiriin. Nämä ovat taideperustaiselle tutkimukselle yleisiä tutkimusaiheita (Jokela & Huhmarniemi 2020, 42).

Taideperustaisessa toimintatutkimuksessa tutkimus etenee usein syklisesti samoin kuin toimintatutkimuskin eli suunnittelun, toiminnan, havainnon ja reflektion kautta (Jokela & Huhmarniemi 2020, 45; Niemi, Kiilakokski & Kaukko 2023, 4:4). Toimintatutkimuksesta poiketen taideperustaiseen toimintatutkimukseen sisältyvät myös taiteen toimintatavat, joita sovelletaan tutkimusaineiston keräämiseen tai analysointiin (Jokela & Huhmarniemi 2020 41–42). Tässä tutkimuksessa luova ohjelmointi ei ollut pelkästään opetuksen sisältö, vaan myös taiteellisen työskentelyn materiaali ja väline. Lukio-opiskelijoiden oppiminen tapahtui luovan ohjelmoinnin avulla taiteellisen työskentelyn kautta.

Kiinnostukseni ei koske ainoastaan luovaa ohjelmointia vain opetuksen näkökulmasta vaan myös sitä millaisia kokemuksia lukio-opiskelijat rakensivat luovan ohjelmoinnin kokonaisuuden seurauksena. Lukio-opiskelijoiden kokemukset eivät kertoneet ainoastaan siitä, miten kokonaisuus toimi vaan myös siitä, millaisena se koettiin. Eli toisin sanoen, miksi luovaa ohjelmointia kannattaa toteuttaa lukion kuvataiteen opetuksessa. Hypoteesina ajattelin, että luovan ohjelmoinnin kokonaisuus voisi paitsi kehittää opiskelijoiden kykyä ymmärtää ohjelmointia myös herätellä ajattelemaan kriittisesti teknologiaa ja digitaalista ympäristöä. Taideperustainen toimintatutkimus tähtää usein hyvinvointia, elinvoimaa sekä työ- ja toimintakykyä kohottavaan toimintaan (Jokela & Huhmarniemi 2020, 46). Tutkimuksen tavoitteena oli kehittää tapa, jolla luovaa ohjelmointia voitaisiin opettaa osana lukion kuvataiteen opetusta. Se ei siis ole ainoa oikea tapa, jolla sitä tulisi opettaa vaan lähinnä esimerkki ja ehdotus, jota voisi tulevaisuudessa kehittää paremmaksi ja toimivammaksi. Taideperustaiselle toimintatutkimukselle on tyypillistä, että tuloksena on kehittövä ja sovellettava esimerkki pysyvän toimintatavan sijaan (Jokela & Huhmarniemi 2025, 107).

Taideperustaisessa toimintatutkimuksessa aloitetaan toimintatutkimuksen tapaan alkukartoituksella (Hakala 2024, 94; Jokela & Huhmarniemi 2025, 115). Tavoitteena oli jäsentää tutkimuksen pedagoginen ja sisällöllinen lähtökohta. Olin jo jonkin aikaa ollut kiinnostunut teknologian (lähinnä ohjelmoinnin) hyödyntämisestä luovassa työskentelyssä, mutta heräte tähän tapahtui Lapin yliopiston kuvataidekasvatuksen Mediataide, pelillisuus ja ohjelmointi kurssilla, jossa tutustuin luovaan ohjelmointiin käsitteenä. Tästä alkoi kartoitus koskien sitä, miten ohjelmointia opetetaan kuvataiteen opetuksen yhteydessä, opetetaanko sitä ja kuinka laaja ilmiö on kyseessä. Alkukartoituksessa tarkastelin myös aiempaa tutkimusta ohjelmoinnin opetuksesta kuvataiteen kontekstissa. Ranta (2020) on tutkinut pro gradussaan peruskoulun kuvataideopettajien kokemuksia ohjelmoinnin hyödyntämisessä ja miten he ovat sitä hyödyntäneet. Nojonen (2024) on tarkastellut pro gradussaan ohjelmoinnin opetuksen laajuutta lukiossa ja sen kytkeytymistä oppiaineisiin. Dufva (2018) on kirjoittanut väitöskirjan luovaa ohjelmointia koskien. Näiden perusteella muodostui kuva siitä, että ohjelmointia on jonkin verran integroitu kuvataiteen opetukseen, mutta etenkin lukion kuvataideopetuksen kontekstissa luovaa ohjelmointia käsittelevää tutkimusta oli niukasti. Tämän pohjalta tarkentui tutkimuksen tavoite kehittää luovaan ohjelmointiin perustuva kokonaisuus, jonka voisi perustellusti integroida lukion kuvataiteen opetukseen.

Laadin suunnitteluvaiheessa tuntikokonaisuuden, jonka tavoitteena oli integroida luova ohjelmointi osaksi lukion kuvataiteen opetusta. Suunnittelua ohjasi opintojakson rakenne, jossa tuntikokonaisuuteen käytettävä aika tuli olla tasapainossa muun opintojakson materiaalin ja tehtävien kanssa. Huomioitavaa oli myös, että tehtävät eivät olleet liian raskaita tai aineiston rakentuminen ei olisi lukio-opiskelijan näkökulmasta suhteettoman raskas (Niemi, Kiilakoski & Kaukko 2023, 4:4). Tutustuin olemassa olevaan materiaaliin ja sen hyödyntämisen mahdollisuuksiin, kun suunnittelin luovan ohjelmoinnin kokonaisuutta. Esimerkiksi Mehackit-sivustolla on selkeä ja laaja kokonaisuus koskien luovaa ohjelmointia (Mehackit n.d.) ja myös matemaattisten aineiden opettajien liiton sivuilla on opetuskokonaisuus otsikolla ”Taidetta ohjelmoimalla” (MAOL n.d.), mutta molemmissa ongelma oli se, että ne olivat aivan liian laajoja ja syvälle meneviä siihen nähden, kuinka paljon aikaa minulla olisi käytettävänä. Hyödynsin edellä mainittujen esimerkkien teemarakennetta, mutta omani oli paljon tiivistetympi eikä lähtenyt sukeltamaan aiheissa niin syvälle. Halusin sisällyttää luovan ohjelmoinnin kokonaisuuteen vahvemmin lukio-opiskelijan pohdintaa digitalisuuteen ja teknologiaan liittyen, joten pohdittavat aiheet määräytyivät tuntien teemojen mukaan. Koottuani tuntimateriaalin, tarkastutin ajatukseni vielä tieto- ja viestintätieteiden alaa opiskelevalla henkilöllä, jotta käsitteet ja sisällöt pysyisivät ymmärrettävinä.

Toteutus on osa toista sykliä, jossa yhdistyy usein myös kolmas sykli eli toiminta ja aineiston rakentuminen taiteellisen toiminnan seurauksena (Jokela & Huhmarniemi 2020, 45; 2025, 115).

Tuntikokonaisuus toteutettiin lukion kuvataiteen opintojakson yhteydessä. Toteutuksen aikana jouduin muokkaamaan materiaalia tilanteen mukaan. Tämän vuoksi jotkin sisällöt etenivät suunniteltua nopeammin. Esimerkiksi värien määrittelyyn ei syvennytty alkuperäisen suunnitelman mukaan. Lisäksi opetuksen aikana lisäsin yhden esimerkin, jossa tarkasteltiin valokuvan hyödyntämistä osana ohjelmointia. Valokuvan lisääminen lähti yhden opiskelijan halusta yhdistää valokuvaa omaan projektiinsa, jonka vuoksi tein siitä kaikille yhteisen esimerkin. Tämän tyyppinen työskentely on ominaista taideperustaiselle toimintatutkimukselle, jossa toiminnan aikana arvioidaan prosessin toimivuutta ja korjataan sitä (Jokela & Huhmarniemi 2025, 113). Tämän jälkeen tuli reflektiovaihe, johon kuuluu aineiston analysointi, käsitteenmuodostus ja seuraavan syklin valmistelu tulosten pohjalta (Jokela & Huhmarniemi 2020, 45; 2025, 115). Alun perin luovan ohjelmoinnin opetus oli tarkoitus järjestää kaksi kertaa, jotta olisin ehtinyt toteuttamaan syklin toiseen kertaan. Ensimmäinen sykli syksyn ensimmäisen opetusharjoittelun aikana ja toinen sykli syksyn toisen opetusharjoittelun aikana. Tämän esti se, ettei opiskelijoiden tietokoneille saanut ladata Processing -ohjelmaa. Tieto kulki todella hitaasti enkä ehtinyt keksiä vaihtoehtoja ratkaisua ennen opetusharjoittelun loppua.

## 2.2 Aineisto ja sen analyysi

Tutkimusaineisto muodostui useista opetuskokonaisuuden aikana syntyneistä materiaaleista. Taideperustaisessa toimintatutkimuksessa aineisto rakentuu muun muassa taideteoksista, tutkijan muistiinpanoista ja toiminnan aikana syntyneistä visuaalisista tuotoksista (Jokela & Huhmarniemi 2020, 50). Aineistoon kuuluivat opiskelijoiden kirjalliset reflektiotekstit ja p5.js-ympäristössä toteuttamat harjoitustyöt ja lopputeokset, opetustilanteissa tehdyt havainnot, alku- ja loppukyselyt sekä opetuskokonaisuuden suunnittelu- ja opetusmateriaalit. Reflektiotekstejä oli 19, harjoitustehtäviä ja itsenäisiä teoksia oli yhteensä 67. Alkukyselyyn vastasi 21 ja loppukyselyyn 15. Opetusmateriaalina olin tehnyt diaesityksen, jossa dioja oli 23 ja esimerkkitehtäviä 6. Lisäksi mainitsen tekstissäni Padlet-alustalla käydyn yhteisen keskustelun, mutta koska tämän tutkimuksen aineistona käyty tuntikokonaisuus oli osa opintojaksoa eikä kaikki keskusteluun osallistuvat olleet antaneet tutkimuslupaa, niin en voi analysoida käytyjen keskustelujen sisältöä tarkemmin vaan viittaan yleisesti omaan tuntemukseeni keskustelun tasosta. Esimerkiksi: ”Tällä tunnilla erityisen hedelmällistä oli opiskelijoiden Padlet-keskustelu” (kts. s. 30). Keskustelua herättelevät kysymykset (LIITE 1) olin laatinut niin, että ne tukivat tunnin aihetta.

Tutkimuskysymykseni muovaantui tarkemmaksi vasta edettyäni aineiston parissa. Tämä ei ole poikkeuksellista taideperustaiselle toimintatutkimukselle vaan kuuluu pikemminkin sen luonteeseen, jossa ymmärrys tarkentuu tutkimuksen edetessä (Jokela & Huhmarniemi 2020, 49). Minun

täytyi ikään kuin oppia ymmärtämään aineiston ääntä ja mitä se oikeasti halusi kertoa, eikä vain sitä mitä itse ajattelin sen kertovan. Tutkimuksen myöhäisessä vaiheessa tajusin, että aineisto halusi tuoda voimakkaasti esiin sen, miksi luovan ohjelmoinnin opettaminen oli merkityksellistä. Tämän seurauksena muokkasin tutkimuskysymystä, jotta se vastasi myös tähän havaintoon. Tähän havaintoon en kuitenkaan päätenyt yksin vaan se vaati keskustelua graduseminaareissa yhdessä ohjaajani ja opponenttien kanssa.

Aineiston analyysin aikana keskustelin aineistoni ja teoriakirjallisuuden välillä. Tämäkin on tyypillistä eikä mitenkään poikkeuksellista taideperustaisessa toimintatutkimuksessa (Jokela & Huhmarniemi 2020, 52). Käytin hyväkseni Dufvan (2018) havaintoja ja väittämiä väitöskirjassaan siitä, että luova ohjelmointi luo ymmärrystä ja omakohtaista tietoa digitaalisuutta kohtaan, kun sitä käytetään taidekasvatuksessa. Samalla Dufva (2018) näkee luovan ohjelmoinnin kokemuksellisenä prosessina, jonka seurauksena syntyy luovaa ja kriittistä ajattelua, mutta samaan aikaan luova ohjelmointi vaatii myös näitä asioita. Knochelin & Pattonin (2015) ajatukset siitä, että ohjelmoinnillisen ajattelun opettaminen kehittää teknologian kriittistä tarkastelua ohjasivat myös ajatteluani. Tämän seurauksena keskityin analyysissä erityisesti siihen, miten lukio-opiskelijat puhuivat teoksien sisällöistä ja visuaalisista ratkaisuista sekä millaisena ohjelmointi koettiin ja minkälaista laajempaa ymmärrystä luovan ohjelmoinnin kokonaisuus herätti. Tekemääni analyysia voisi näin ollen luonnehtia dialogiseksi temaattiseksi analyysiksi (Koski 2020).

Tutkimuksen varsinainen analyysi rajattiin ensisijaisesti opiskelijoiden kirjallisiin reflektioteksteihin. Rajaus tehtiin, koska reflektiotekstit tarjosivat suoran näkymän siihen, miten opiskelijat jäsentivät kokemuksiaan, oppimistaan ja luovaa työskentelyään. Muu aineisto toimii analyysia täydentävänä ja tulkintaa tukevana materiaalina. Opiskelijoiden teokset olivat tutkimuksessa merkityksellisenä osana oppimisprosessia ja ilmaisullista työskentelyä, mutta niitä ei analysoitu erillisenä visuaalisena aineistona systemaattisesti. Niiden tehtävänä oli täydentää ymmärrystä siitä kontekstista, jossa opiskelijoiden reflektiot syntyivät. Kirjalliset pohdinnat tuotettiin apukysymysten avulla, joita opiskelijat saivat halutessaan hyödyntää kirjoittaessaan reflektiota (LIITE 2). Tällöin aineistoa voidaan luonnehtia puolistrukturoiduksi siinä mielessä, että apukysymykset ohjasivat reflektiotekstien sisältöjä, mutta eivät rajoittaneet vastausten muotoa tai rakennetta. Tämä näkyi siinä, että suurin osa opiskelijoista (84,21 %) viittasi apukysymyksiin suoraan (Esimerkki 1) ja loput opiskelijoista (15,79 %) viittasivat apukysymyksiin epäsuorasti osana reflektiotaan (Esimerkki 2). Osa vastaajista eteni selkeästi kysymykohtaisesti (Esimerkki 1), kun taas osa kirjoitti vapaamuotoisempaa reflektiota, jossa apukysymykset toimivat taustalla ohjaavana rakenteena ilman tavanomaista kysymys-vastaus-asetelmaa (Esimerkki 2).

(1) Esimerkki:

Millaista oli tehdä taidetta koodin avulla verrattuna perinteisiin menetelmiin? Erittäin mielenkiintoista sekä mukavan vaihtelevaa. -Opiskelija 10

(2) Esimerkki:

Koodin avulla tekeminen oli luonnollista, mutta tässä kohtaa osuin juuri siihen kuoppaan, missä visiota on vähän liikaa, ettäkö siitä erkaantuminen olisi helppoa, mutta liian vähän siihen nähden, että työ pitäisi saada valmiiksi rajallisessa ajassa. Ehkä paremman, yhden yhtenäisen lopputuloksen olisi saanut aikaan valitsemalla yhden idean, suunnittelemalla sen hyvin, ja toteuttamalla vasta sitten. -Opiskelija 1

Aineiston sisäinen vaihtelu näkyy erityisesti siinä, miten opiskelijat jäsensivät kokemuksiaan ja ajatteluaan suhteessa tehtävänannon ohjaukseen. Esimerkit 1 ja 2 havainnollistavat näitä erilaisia vastaustapoja. Ensimmäisessä esimerkissä opiskelija vastaa suoraan apukysymykseen ja toisessa esimerkissä reflektio rakentuu laajemmaksi prosessikuvaukseksi, jossa opiskelija tarkastelee työskentelyään, päätöksentekoaan ja ajattelunsa kehittymistä ilman suoraa viittausta yksittäiseen kysymykseen.

Analyysia tehtiin teoriaohjaavasti. Dufvan (2018) ja Knochel & Pattonin (2015) ajatukset ohjasivat osin luovan ohjelmoinnin tuntikokonaisuuden rakentumista. Keskityin kokonaisuudessa antamaan opiskelijoille uuden työkalun ja materiaalin, jonka avulla voisi lähteä ilmaisemaan itseään. Reflektiotehtävän olin suunnitellut osaksi Koodattu maailma -tehtävää. Opintojakson aikaisemmat tehtävät olin rakentanut samalla periaatteella eli opiskelijoiden tuli kertoa prosessistaan ja ajatuksistaan työn takana kirjallisesti, joten myös luovan ohjelmoinnin kokonaisuuden tehtävä noudatti samaa kaavaa.

Teemoittelussa aineistosta pyritään tunnistamaan aineiston kannalta keskeisiä aihepiirejä (Hakala 2024, 102), kun taas dialogisessa tematisoinnissa teoreettinen ajattelu on läsnä ja luo tilan, jossa teoria on analyysin keskustelukumppani (Koski 2020). Huomioitavaa on, että teemat sisältävät aina myös tutkijan tulkintaa siitä, mikä on tutkimusongelman näkökulmasta olennaista (Vuori 2021). Analysoin aineistoani dialogisen tematisoinnin kautta. Ilmiötä lähdetään tarkastelemaan yleisen ja erityisen kontekstin kautta (Koski 2020). Tässä tutkimuksessa yleisenä kontekstina toimii luova ohjelmointi. Se miten se ilmenee, miten se käsitetään ja miten sitä on käytetty. Yleinen konteksti siis muodostuu teorian pohjalta, siitä miten ilmiö näyttäytyy (Koski 2020). Eri-tyinen konteksti taas on tässä tutkimuksessa lukion nykytaiteen opintojakso. Eli se, missä tutkittava ilmiö sijaitsee (Koski 2020). Kosken (2020) mukaan dialogisessa teemoittelussa voidaan muokata alkuperäisiä teoreettisia lähtökohtia niin, että niiden avulla voidaan tulkita havaintoja. Eli teoriaa ei lyödä heti samalta istumalta lukkoon vaan se kehittyy tutkimuksen edetessä. Tämä

sopii vahvasti taideperustaisen toimintatutkimuksen ajatuksen kanssa, jossa tutkijan ymmärrys tarkentuu analysoinnin edetessä (Jokela & Huhmarniemi 2020, 49). Teoria muovaantui analyysin edetessä. Huomasin kaipaavani erityisesti taiteellisesta ilmaisusta enemmän tietoa sekä ohjelmointiin liittyen, joten erityisesti nämä teoriat elivät ja tarkentuivat analyysin edetessä.



Kuva 1. Ajatuskartan hahmottelua. Jenna Hepoaho 2026.

Analyysin ensimmäisessä vaiheessa tutustutaan aineistoon ja tarkastellaan sitä teoreettisen ajattelun ja tutkimuskysymyksen näkökulmasta (Koski 2020). Tässä vaiheessa jäsenyivät kaksi pääteemaa. Nämä teemat perustuivat luovan ohjelmoinnin kokonaisuuden kahteen keskeiseen ulottuvuuteen luovuuteen ja ohjelmointiin. Eli taiteelliseen työskentelyyn ja ohjelmointiin sen välineenä ja materiaalina. Aloin tarkastella tästä näkökulmasta opiskelijoiden ilmauksia liittyen luovuuteen ja ohjelmointiin. Toisessa vaiheessa aineistosta etsitään toistuvia, perusteltuja tai painotettuja asioita (Koski 2020). Hyödynsin ajatuskarttaa, johon pelkistin opiskelijoiden ajatukset ja loin kategorioita (Kuva 1). Esimerkkiotteita opiskelijoiden reflektioteksteistä, jotka sijoittuvat ohjelmointi -ajatuskarttaan. Kaikki teemat rakentuivat tällä samalla logiikalla. Teorian kanssa puhuminen ilmeni tässä tapauksessa esimerkiksi niin, että pystyin havainnoimaan analyysiä tarkemmin. Esimerkiksi Opiskelija 11 ilmaisee ymmärtäneensä kokonaisuuden jälkeen koodausprosessia. Teorian valossa sanallistin tämän koodin lukutaitoon ja ohjelmoinnilliseen ajatteluun.

Taiteen tekeminen koodin avulla on mielestäni haastavaa, sillä koodeissa on niin paljon sääntöjä ja rajoitteita, joita on vaikea hahmottaa. -Opiskelija 4

Koodaaminen antoi minulle uusia mahdollisuuksia laajentaa ymmärrystä ja soveltaa omaa osaamistani kuvataiteessa. -Opiskelija 8

Oikeastaan koko koodausprosessia, että miten se tapahtuu. Myös sen, että pienikin yksityiskohta vaikuttaa. -Opiskelija 11

Että pystyn koodaamalla luomaan paljon erilaisia teoksia, joissa voin ilmaista itseäni. -Opiskelija 12

Koodaaminen voi olla yhtä luovaa kuin mikä tahansa muu taiteen laji. -Opiskelija 15

Turhauttavaa kun ei osaa. -Opiskelija 18

Koodi on tapa ilmaista itseään ja luoda esteettisiä kokemuksia ruudun välityksellä. -Opiskelija 19

Analyysin edetessä aineistosta nousi havaintoja, jotka eivät mahtuneet ilmaisun tai ohjelmoinnin alle. Tässä vaiheessa aineisto nosti esiin myös oppimiskokemuksia, pystyvyyden tuntemuksia, ajattelun muutoksia ja taidekäsitteitä. Teema kulki nimellä ”laajemmat kokonaisuudet”. Kolmannessa vaiheessa muodostetaan teemahypoteesit, jotka testataan eli aineisto käydään uudestaan läpi näiden teemojen valossa (Koski 2020). Tässä vaiheessa tarkentuivat lopulliset teemat, jotka nimesin Taiteellinen ilmaisu, Ohjelmointi ja Erityiset merkitykset. Näistä muodostui myös tämän tutkimuksen tulokset.

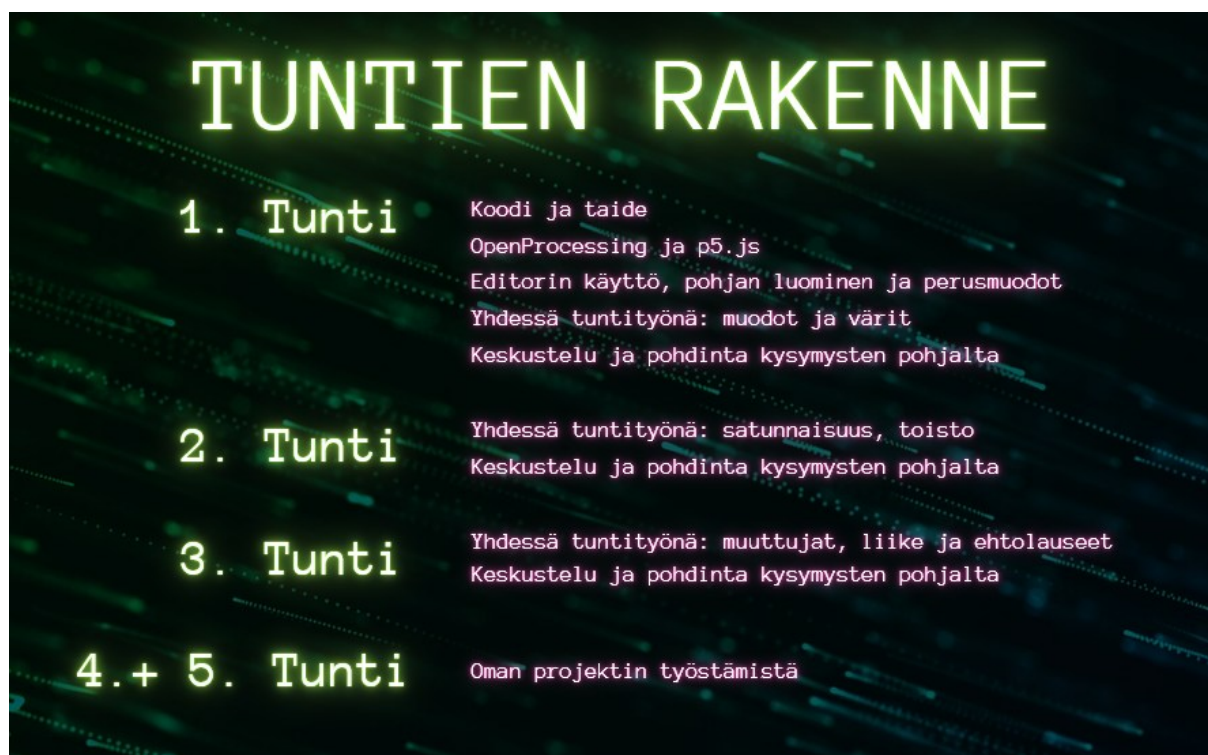
### 2.3 Tutkimuksen eettisyys

Tutkimuksessa noudatettiin hyvän tieteellisen käytännön periaatteita. Hallinnollisissa instituutioissa, kuten esimerkiksi lukiossa vaatii omat lupansa (Jokela & Huhmarniemi 2020, 54). Näin olen tutkimuslupa anottiin kaupungilta sekä lukion rehtorilta. Opiskelijoiden osallistuminen tutkimukseen perustui asianmukaisesti tutkimuslupiin (LIITE 3) ja aineisto käsiteltiin anonymisoidusti siten, ettei yksittäisiä osallistujia voida tunnistaa raportista. Tutkimuslupa sisällytti myös lupa käyttää opiskelijoiden töitä osana aineistoa ja esittämistä, mikä myös tulee huomioida taideperustaisessa toimintatutkimuksessa (Jokela & Huhmarniemi 2020, 54). Taideperustaisessa toimintatutkimuksessa tutkijan osallistuva rooli vaikuttaa väistämättä tutkimusprosessiin. Tässä tutkimuksessa tutkija toimi samanaikaisesti opettajana ja aineiston tulkitsijana, mikä on huomioitu tutkimuksen luotettavuuden arvioinnissa. Tutkijan läheinen suhde tutkimuskontekstiin mahdollisti syvällisen ymmärryksen opetustilanteista, mutta samalla se edellytti kriittistä reflektiota omista tulkinnoista. Tutkimuksen tavoitteena ei ollut tuottaa yleistäviä tuloksia kaikista lukioympäristöistä vaan kuvata ja ymmärtää yhden opetuskokeilun pohjalta, millaisia mahdollisuuksia luovalla ohjelmoinnilla voi olla osana kuvataiteen opetusta.

Toimin tutkijana, mutta myös opettajana suorittaessani samalla syventävää harjoittelua ja vastaisin tuntikokonaisuuden suunnittelusta, toteutuksesta ja aineiston tulkinnasta. Tällainen kaksoisrooli on taideperustaiselle toimintatutkimukselle tyypillinen (Jokela & Huhmarniemi 2020, 47). Opettaessani koko opintojakson enkä pitänyt ainoastaan yksittäistä opetuskokeilua pystyin liittämään luovan ohjelmoinnin osaksi opetuksen sisältöä luontevasti. Tämä mahdollisti myös opiskelijoille luontevan siirtymän kohti ohjelmointia ilman, että aihe näyttäytyi täysin irrallisena. Opetustilanteisiin liittyvät havainnot kirjattiin tuntien jälkeen muistinvaraisesti, mikä on voinut vaikuttaa havaintojen tarkkuuteen ja painotuksiin. Luovan ohjelmoinnin osuus opintojaksosta oli sen lopussa, jolloin opiskelijat olivat jo ehtineet tutustua minuun ja minä heihin eli jonkin asteinen luottamus oli ehtinyt syntyä.

### 3 Luovan ohjelmoinnin opetuskokonaisuus kuvataiteen opetuksessa

Tässä luvussa kuvataan tutkimuksen ytimessä olevaa opetuskokonaisuutta, jossa luova ohjelmointi tuotiin osaksi kuvataiteen opetusta nykyaikaisen opintojaksolla. Kokonaisuus muodostuu sekä suunnitteluvaiheesta että sen käytännön toteuttamisesta opetustilanteessa. Koska kyseessä on taideperustainen toimintatutkimus, opetuskokonaisuuden rakenne ei ole vain osa taustaa vaan se on myös keskeinen osa tutkimuksen tuloksia.



| TUNTIEN RAKENNE |   |
|-----------------|---|
| 1. Tunti        | Koodi ja taide<br>OpenProcessing ja p5.js<br>Editorin käyttö, pohjan luominen ja perusmuodot<br>Yhdessä tuntityönä: muodot ja värit<br>Keskustelu ja pohdinta kysymysten pohjalta |
| 2. Tunti        | Yhdessä tuntityönä: satunnaisuus, toisto<br>Keskustelu ja pohdinta kysymysten pohjalta  |
| 3. Tunti        | Yhdessä tuntityönä: muuttujat, liike ja ehtolauseet<br>Keskustelu ja pohdinta kysymysten pohjalta   |
| 4.+ 5. Tunti    | Oman projektin työstämistä  |

Kuva 2: Kuvankaappaus tuntikokonaisuuden diasta: Tuntien rakenne.

Luovan ohjelmoinnin tuntirakenne (Kuva 2) eteni vaiheittain kohti haastavampia sisältöjä. Jokainen tunti sisälsi yhdessä kirjoitetun esimerkkikoodin ja sen muokkaamisen sekä tunnin lopussa käytävän yhteisen pohdinnan valmiiksi laatimieni kysymysten pohjalta (LIITE2). Kolme ensimmäistä oppituntia olin varannut ohjelmoinnin perusteiden hallintaan ja viimeiset kaksi oppituntia lukio-opiskelijoiden itsenäisen teoksen työstämiseen. Luvussa 3.2 käyn läpi tarkemmin jokaisen tunnin sisällön ja etenemisen.

Luovan ohjelmoinnin opetuskokonaisuus asettui rajatuksi, mutta intensiiviseksi kokeiluksi, jossa kolme 75 minuutin oppituntia ja kaksi työskentelytuntia rakensivat opiskelijoille kosketuksen koodiin kuvataiteen kontekstissa. Rakenne tuki erityisesti sitä, että lukio-opiskelijat uskalsivat kokeilla ja muokata yhteisiä esimerkkikoodeja, mutta samalla se rajasi pois syvällisemmän perehtymisen esimerkiksi generatiiviseen taiteeseen ja monimutkaisempiin ohjelmointirakenteisiin.

### **3.1 Suunnittelun lähtökohdat ja tavoitteet**

Luovan ohjelmoinnin opetuskokonaisuuden suunnittelu lähti liikkeelle tarpeesta kokeilla ja tarkastella, millä tavoin luovaa ohjelmointia olisi mahdollista integroida lukion kuvataiteen opetukseen. Tavoitteena ei ollut rakentaa kattavaa yhden opintojakson ohjelmoinnin opetuskokonaisuutta vaan hyvin rajattu kokonaisuus osana Nykytaiteen opintojakson sisältöä, jonka kautta olisi mahdollista tarkastella, miten ohjelmointia voisi tuoda mukaan lukion kuvataiteen opetukseen. Huomion kohteena eivät olleet ensisijaisesti teknilliset taidot sinänsä, vaan se, millaisia ymmärryksen muotoja ja ajattelun tapoja luovan ohjelmoinnin työskentely voi synnyttää. Opetuskonteksti asettaa suunnittelulle selkeät reunaehdot. Kokonaisuus toteutettiin osana lukion nykytaiteen opintojaksoa, mikä tarkoitti, että luova ohjelmointi ei voinut muodostaa koko opintojakson sisältöä vaan sen tuli asettua yhdeksi tehtäväkokonaisuudeksi muiden sisältöjen rinnalle. Käytävissä oleva aika oli rajallinen ja samaan aikaan opetuksen tuli olla saavutettavaa opiskelijoilla, joilla ei ollut aiempaa kokemusta koodaamisesta.

Teoreettinen viitekehys oli suunnitteluvaiheessa vielä osin hahmottumassa, mutta sitä ohjasivat vahvasti aiemmat havainnot luovasta ohjelmoinnista pedagogisena ja kulttuurisena ilmiönä. Eriytisesti Taiteen tohtori Tomi Dufvan (nykyisin Slotte Dufva) väitöskirjan (2018) sekä Filosofian tohtorien Christine Liao Knochel ja Ryan M. Pattonin (2015) kirjoitukset tarjosivat näkökulmia koodin ymmärtämiseen merkityksiä tuottavana ja kokemuksellisena toimintana, ei vain teknisenä välineenä. Näiden lähtökohtien pohjalta tuntirungon ja materiaalin suunnittelu rakentui ajatukselle koodista osana nykytaiteellista ajattelua ja post-digitaalisen maailman kriittistä tarkastelua.

Tuntirunko rakentui opetuksen osalta kolmen 75 minuutin oppitunnin kokonaisuudeksi, jossa sisällöt rajattiin tietoisesti peruskäsitteisiin ja -toimintoihin. Lisäksi opiskelijoille oli varattu kaksi oppituntia itsenäisen teoksen valmistamiseen. Ensimmäisellä tunnilla keskityttiin koodin peruslogiikkaan, selaimessa toimivan alustan käyttöön sekä koodin rakenteen hahmottamiseen. Toisella tunnilla käsiteltiin toistorakenteita ja satunnaisuutta ja kolmannella tunnilla interaktiivisuutta, kuten hiiren ja näppäimistön käyttöä. Lisäksi käsiteltiin lyhyesti kuvan lisäämistä osaksi teosta. Rakenteellinen vaiheistus perustui ajatukseen siitä, että jokainen uusi sisältö rakentuu edellisen varaan ja samalla avaa uusia ilmaisullisia mahdollisuuksia.

Tutkimuksen kuvallinen aineisto koostuu opiskelijoiden p5.js:llä toteuttamista töistä. Vaikka p5.js:stä puhun opetustilanteissa siitä ohjelmointikielenä, se on teknisesti ottaen JavaScriptin päälle rakennettu kirjasto, joka laajentaa JavaScriptin toiminnallisuuksia ja tuo siihen Processing-ympäristöstä tuttuja visuaalisen ohjelmoinnin piirteitä. Samoin kuin Processing, myös p5.js on kehitetty ohjelmoinnin ja taiteen oppimista varten (p5.js 2026). Alun perin ajattelin käyttää ohjelmointiympäristönä ja -kielenä Processingia, mutta tiukentuneiden käyttörajoitusten vuoksi opiskelijoiden koneille ei saanut kyseistä ohjelmaa ladata.

Työskentely toteutettiin reaaliaikaisen ohjelmoinnin (live coding) -menetelmää hyödyntäen. Tässä menetelmässä opettaja tuottaa koodia reaaliaikaisesti opiskelijoiden nähden ja sanallistaa samalla ohjelmoinnin prosessia. Menetelmä on vakiintunut ohjelmoinnin opetuksessa, koska se tekee asiantuntijan ajattelun näkyväksi ja tukee opiskelijoiden vaiheittaista oppimista. (Brown & Wilson 2019, 3; Fuentes Martinez 2024.) Tunnin aikana lukio-opiskelijat kirjoittavat yhdessä opettajajohtoisesti saman esimerkkikoodin, jota myöhemmin lähdetään itse soveltamaan. Esimerkkikoodi oli jaettu lukio-opiskelijoille Pedanettiin, jota he olivat tottuneet käyttämään. Tallennettu esimerkkikoodi oli varustettu kommentteilla, jotta siihen pystyisi palaamaan jälkikäteen ja ymmärtämään mistä on kyse (Kuva 3). Sen oli myös tarkoitus tukea koodin lukemisen ja ymmärtämisen taitoa.

```

mySketch
1  /*Tässä harjoituksessa harjoitellaan muotojen piirtämistä pohjalle ja värien lisäämistä muotoihin
2  Värien hakeminen ja suunnitteleminen on helppoa hyödyntäen esim. Adobe Color -sivutoa.
3  Ohjelma käyttää myös RGB värejä = (R,G,B) harmaa-asteikko = 0-255*/
4
5  //setup = pohjan muuttumaton paperi eli ohjelma ajaa tämän sisällön kerran ylhäältä alaspäin
6  function setup() {
7    createCanvas(400, 400); // pohjan koko (pikseliä)
8    background(100); // pohjan tausta: harmaa 0=musta, 255=valkoinen
9  }
10
11 //tähän kaikki minkä haluat näkyvän koko ajan. Ohjelma ajaa tätä looppina eli pysähtymättä 60x sekunnissa ylhäältä alaspäin
12 function draw() {
13
14   fill('#E37862')
15   noStroke() // poistaa ääriviivat
16   circle(150, 67, 80); //ympyrä sijainti x-akselilla, sijainti y-akselilla ja halkaisijan koko
17
18   fill('#43D9A6')
19   ellipse(334, 100, 50, 100); //ellipsin sijainti x-akselilla, y-akselilla ja leveys, korkeus
20
21   stroke('#DBF227') //ääriviivan väri
22   strokeWeight(5); // lisätään ääriviivan paksuutta, jotta piste näkyy
23   point(67, 200); //piste
24   strokeWeight(1); //palautetaan viivan paksuus yhteen
25   stroke(0)//ääriviivan väri musta
26
27   line(175,225, 225, 175); //viiva. Ensimmäinen piste x, y ja toinen piste x, y. Yhdistää pisteet viivalla
28
29   quad(260, 300, 260, 200, 300, 180, 300, 260);// Yhdistää neljä pistettä viivalla.
30
31   rect(42, 314, 50, 40);//suorakulmio
32
33   square(150, 250, 100);//neliö. Sama kuin ympyrässä eli kokoon riittää yksi luku
34
35   triangle(315, 340, 342, 310, 350, 355); // kolmio
36 }

```

Kuva 3: Esimerkkikoodi kommenttien kanssa.

Kommentit olivat sisällöltään sama mitä tunnilla yhteistä esimerkkiä tehdessä kerroin. Tämä ratkaisu pyrki madaltamaan kynnystä koodin pariin ja tukemaan onnistumisen kokemuksia erityisesti niille opiskelijoille, joille ohjelmointi oli täysin vierasta.

Yleisiksi oppimistavoitteiksi asetin, että lukio-opiskelija kykenee hahmottamaan koodin perusrakennetta, tuottamaan itsenäisesti luovan teoksen opetuskokonaisuuden jälkeen sekä pohtimaan kriittisesti digitaalista maailmaa. Näitä tavoitteita tuettiin tuntien lopussa esitetyillä reflektiivisillä kysymyksillä, jotka laajensivat teknistä työskentelyä käsitteelliselle tasolle (LIITE 2). Kysymykset saivat lukio-opiskelijat pohtimaan, miten heihin itseensä voidaan pyrkiä vaikuttamaan heidän toimintaansa ohjaamalla. Lukio-opiskelijat pohtivat kysymyksiä yksin tai ryhmissä ja vastaukset koottiin yhteiselle Padlet -alustalle. Näin tuntirunko ei pyrkinyt ainoastaan opettamaan koodaamista, vaan sijoittamaan sen osaksi laajempaa taiteellista ja yhteiskunnallista keskustelua. Opetusmateriaali suunniteltiin tukemaan tuntirungon tavoitteita ja rakennetta. Materiaali koostui harjoitustehtävistä, diaesityksistä sekä lopputehtävästä, jonka otsikkona oli ”koodattu maailma”. Materiaalin rooli oli sekä ohjaava että avaava. Sen tehtävänä oli tarjota perustietoa ja konkreettisia esimerkkejä, mutta samalla jättää tilaa opiskelijoiden omalle kokeilulle ja tulkinnalle.

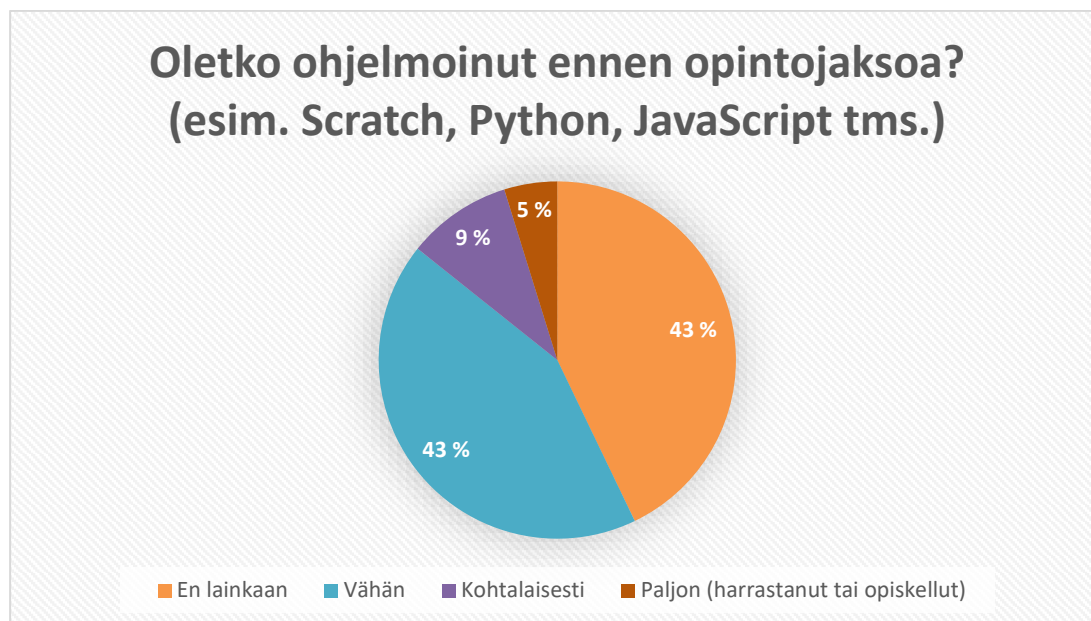
Ohjelmoinnin opettamiseen liittyvä keskeinen haaste liittyi ajankäyttöön. Syvälinen oppimien vaatii aikaa ja ohjelmoinnin kohdalla pienetkin käsitteet voivat vaatia toistoa ja rauhallista etene- mistä. Suunnitteluvaiheessa oli tarpeen tuottaa mahdollisimman kevyt kokonaisuus, joka kuiten- kin mahdollistaisi olennaisten perusasioiden omaksumisen. Verkkoympäristössä saatavilla oleva materiaali osoittautui tässä suhteessa ongelmalliseksi, sillä se oli usein suunnattu pidem- piin opintokokonaisuuksiin tai kytkeytyi vahvasti pitkän matematiikan sisältöihin.

Materiaalissa korostui leikin ja kokeilun ajatus. Tätä ajatusta korostavat myös Knochel & Patton (2015, 25), joiden mukaan koodilla leikkiminen harjoittaa ohjelmoinnillista ajattelua. Opiskelijoi- den tehtävän oli muokata yhteisiä esimerkkikoodeja ja palauttaa oma versionsa tunnin päät- teeksi. Itsenäisesti tuotetut teokset saivat hyödyntää tunneilla käsiteltyjä esimerkkejä, mikä osal- taan hämärsi rajan mallin ja oman työn välillä ja teki näkyväksi koodin avoimen ja muokattavan luonteen. Hypoteesina ajattelin, että tämä ratkaisu voisi tukea sekä teknistä oppimista että kä- sitteellistä pohdintaa tekijyydestä ja omistajuudesta.

Suunnitteluvaiheella oli keskeinen merkitys tutkimuksen toteutumiselle. Opetuskokonaisuus saatiin sovitettua nykyaikaisen opintojakson rakenteeseen, mutta sen rajattu luonne vaikutti myös aineiston laatuun ja määrään. Aineiston keruussa oli huomioitava opiskelijoiden kuormittuminen (Niemi, Kiilakoski & Kaukko 2023, 4:4). Keskeisimmäksi aineistoksi valikoituivat opiskelijoiden it- senäisesti tuottamat teokset sekä niitä ja oppimisprosessia koskeva pohdintateksti, joka oli to- teutettu apukysymyksiä (LIITE 2) käyttäen. Suunnitteluprosessi ei ollut suoraviivainen vaan se ke- hittyi sekä ennako-oletusten että opetustilanteessa syntyneiden havaintojen perusteella. Sa- malla suunnitteluvaihe syvensi omaa ymmärrystä luovasta ohjelmoinnista pedagogisena mene- telmänä. Erityisesti yksinkertaisten, mutta merkityksellisten esimerkkien tuottaminen osoittautui vaativaksi, sillä koodi on usein vaikeasti sanallistettavaa ja omien kokemuksieni mukaan näyttäy- tyy helposti sekavana niille, joilla ei ole aiempaa kokemusta siitä. Tämä havainto on olennainen myös tutkimuksen tulkinnan kannalta, sillä se tekee näkyväksi sen, millaisissa ehdoissa opiske- lijoiden oppimiskokemukset ovat syntyneet.

Alkukysely toteutettiin heti tuntikokonaisuuden ensimmäisellä tunnilla samalla, kun keräsin tut- kimusluvut. Siihen vastasi 21 lukio-opiskelijaa. Sen mukaan suurin osa opintojaksolla mukana olleista opiskelijoista ei ollut koskaan aikaisemmin koodannut tai oli tehnyt sitä vain vähän (Kaa- vio 1). Tämä oli mielenkiintoinen tulos, sillä lukio-opiskelijat ovat suorittaneet suurimmaksi osaksi peruskoulunsa tällä hetkellä olevan opetussuunnitelman mukaan. Nykyinen peruskoulun opetussuunnitelma mainitsee ohjelmoinnin opetuksen laaja-alaisissa tavoitteissa (Opetushalli- tus 2014, 156). Opiskelijoiden kanssa keskusteltaessa asiasta kävi ilmi, että usein ohjelmointi on

koettu olevan aihe, jonka käsittely on jätetty keväälle, jos aika on antanut myöden. Eli se on koettu olevan enemmänkin viihdykettä tai jonkinlaista ”täytettä” viimeisille oppitunneille. Alkukyselyn mukaan 23,8 % oli kiinnostunut tai erittäin kiinnostunut luovasta ohjelmoinnista ilman ennakkotietoja. Digitaalisia välineitä, kuten kuvankäsittelyyn tai videoeditointiin liittyviä ohjelmia ei ollut käyttänyt lainkaan tai vain vähän 57,1 % vastaajista (Kaavio 2). Tästä huolimatta lähes kaikki osallistujat suoriutuivat tehtävistä ja saivat tuotettua oman teoksen oppimansa pohjalta. Poikkeuksena olivat ne opiskelijat, joille kertyi paljon poissaoloja ja suorittivat korvaavan tehtävän.



Kaavio 1. Lukio-opiskelijoiden ohjelmointikokemus.



Kaavio 2. Opiskelijoiden kokemus työskentelystä digitaalisin välinein.

### 3.2 Tuntien eteneminen

Tässä kuvaan, miten tuntikokonaisuus eteni käytännössä. Tämä kuvaus tekee näkyväksi ne pedagogiset ratkaisut, joiden vaikutuksia tarkastelen myöhemmin opiskelijoiden kokemusten ja testosten kautta.

Ennen Koodattu maailma -tuntikokonaisuuden alkamista olimme tutustuneet laajasti opiskelijoiden kanssa nykyaikaiseen ja sen eri ilmenemismuotoihin. Lähdin johdattelemaan opiskelijoita digitaalisen taiteen sekä taidetta ja teknologiaa yhdistäviin teoksiin. Näistä käytin esimerkkinä muun muassa Teija ja Pekka Isorättyän Robohemians (2022) -teosta. Teos oli esimerkkinä osuva, sillä se nousi uutisiin sen jälkeen, kun opintojakson päättymisestä oli mennyt jonkin aikaa. Ennen ohjelmoinnin tuntikokonaisuuden alkamista ohjeistin lukio-opiskelijat luomaan käyttäjätilin openProcessing.org verkkosivustolle. Käytimme tätä alustana ohjelmoinnille, koska se oli mielestäni selkeä ja yksinkertainen. Sinne pystyi myös tallentamaan omat keskeneräiset ja valmiit työt. Vaikka ilmaisversiossa kaikki julkaistu on julkista niin omien testien mukaan sieltä harvoin nousee mitään oikeasti esille. Ainakaan omien testien osalta en saanut ylimääräisiä katselukertoja, kun loin tunneilla käytettävät esimerkit. Mainitsin tästä opiskelijoille ja pyysin huomioimaan, että heidän teoksensa olisivat julkisia.

Ensimmäisellä tunnilla esittelin tuntikokonaisuuden, itsenäisen tehtävän ohjeistuksen ja tavoitteet opiskelijoille. Näytin esimerkkejä ISEA:n (Inter-Society for Electronic Arts) arkistoista sekä esittelin Alain Bittlerin Apohysis -teossarjaa ja Zach Liebermanin yhden nimettömän teoksen. Molemmat esimerkit oli tehty ohjelmoimalla eri ohjelmointiympäristöissä. Tämän jälkeen esittelin lyhyesti ohjelmointikielen, mitä tulisimme käyttämään.

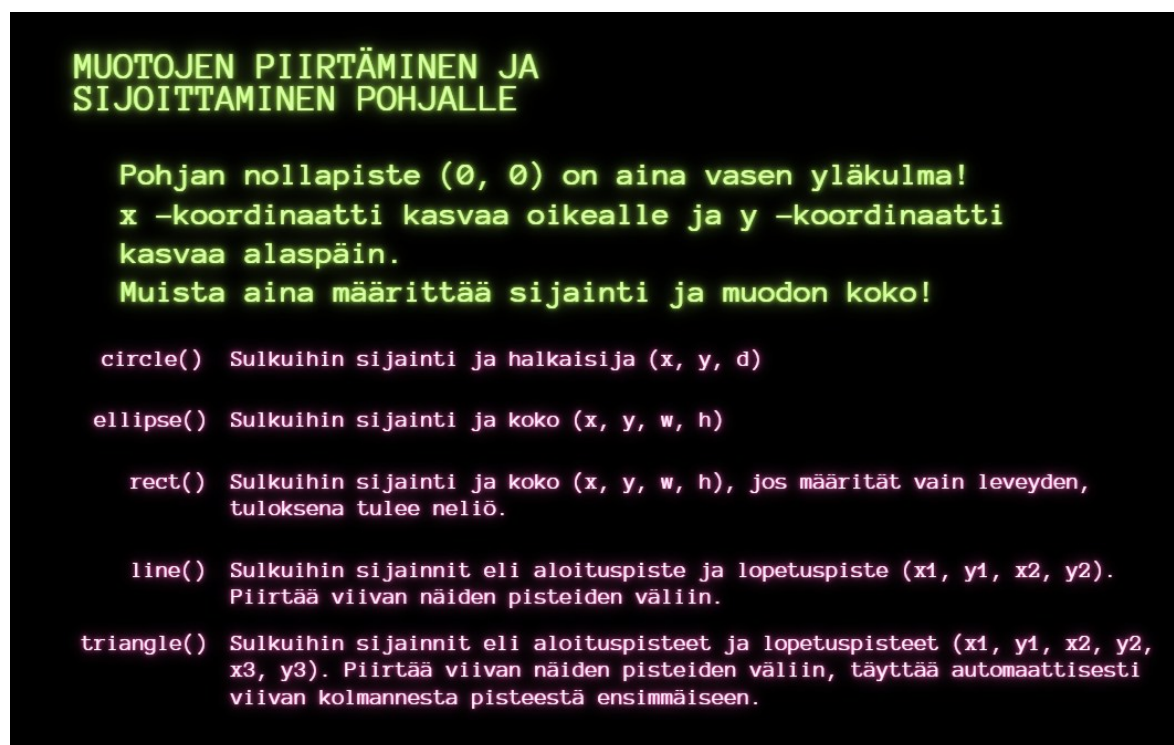
Tehtävänanto opiskelijoille oli seuraava:

“Elämme koodatussa maailmassa. Teknologia ympäröi meitä kaikkialla: puhelimissa, sovelluksissa, tietokoneissa... ja usein niin huomaamattomasti, ettemme edes ajattele sen olemassaoloa. Samalla meissä itsessämme on toinen koodijärjestelmä: geneettinen koodi, joka määrittää solujen toimintaa. Koodi on siis yhtä aikaa sekä digitaalista että biologista – osa arkea ja osa olemusta.

Luo teos, jonka kautta käsittelet koodattua maailmaa. Aihe voi olla mitä tahansa! Voit lähestyä sitä teknologian, biologian tai täysin omasta näkökulmastasi. Hyödynnä tunneilla tehtyjä koodipohjia ja muokkaa niitä oman ideasi mukaan (tämä on jopa suositeltavaa!). Lisäksi p5.js -sivustolta löydät paljon tutoriaaleja, joita voit käyttää inspiraationa ja apuna.”

Tehtävän tarkoituksena oli haastaa opiskelijoita, mutta samaan aikaan olla tarpeeksi lähestyttävä monesta eri näkökulmasta. Ajattelin, että ohjelmointi voi olla monelle haastavaa ja epämu- kavuusalueella vaeltelua, joten pyrin luomaan tehtävänannosta sellaisen, että se sopisi myös tuntitehtäviin.

Ohjelmoinnin osalta ensimmäinen tunti käsitteli tutustumista siihen, miten kirjoitettu koodi toi- mii. Tarkoitus ei ollut lähteä syvällisemmin tutustumaan koodin toimintaan tai sen historiaan vaan niihin ominaisuuksiin, jotka koin merkitykselliseksi ymmärtää, jotta ohjelmoinnissa pääsisi alkuun. Sen jälkeen lähdimme yhdessä opiskelijoiden kanssa tutustumaan logiikkaan käytän- nössä. Lukio-opiskelijat tekivät ohjeistukseni mukaisen pohjan, johon luotiin erilaisia muotoja ja muokattiin muotojen ääriviivoja sekä värejä. Tällä tavoin havainnollistin opiskelijoille, miten tie- tokone ”lukee” koodia aina ylhäältä alaspäin. P5.js on JavaScript -kirjasto, joka tarjoaa valmiiksi määriteltyjä funktioita visuaalista ohjelmointia varten. Ensimmäisellä tunnilla käsitelimme se- tup() -ja draw()-funktioita. Setup()-funktiossa määritellään ohjelman alkuasetukset ja sen sisäl- tämä koodi suoritetaan kerran ohjelman käynnistyessä. Draw()-funktio sisältyvä koodi sen si- jaan toistuu jatkuvasti ohjelman pääsil- mukassa.

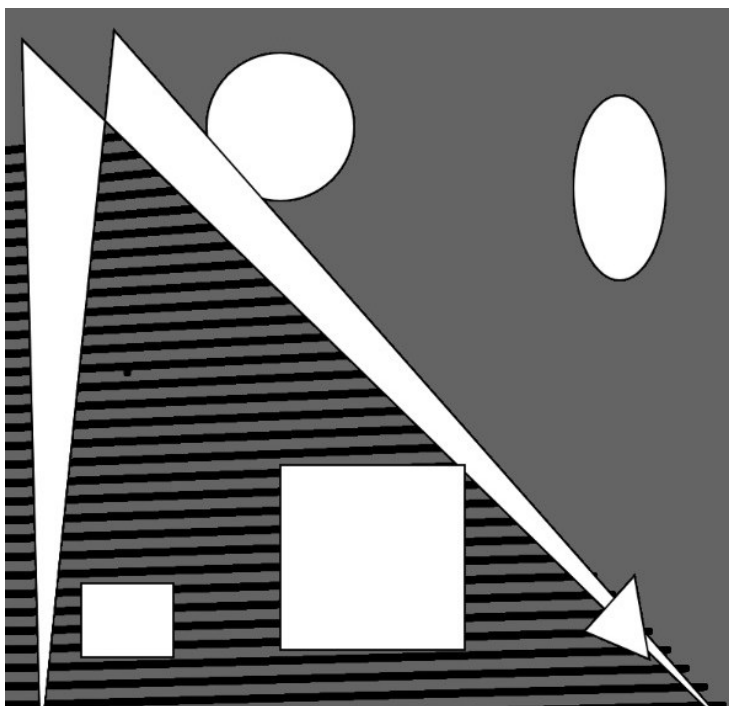


Kuva 4: Esimerkki opiskelijoille jaetuista dioista

Tunnilla yhdessä tehtyjen esimerkkikoodien lisäksi tallensin Pedanettiin valmiit muistiinpanot (Kuva 4), joita lukio-opiskelijat saivat vapaasti hyödyntää. En käynyt näitä dioja tunneilla juurikaan läpi, koska kävimme asiat läpi samalla kun kirjoitimme koodia. Diojen tarkoitus oli olla ”valmiita

muistiinpanoja” opiskelijoille, jotta tunneilla käytyt sisällöt olisivat helposti löydettävissä jälkikäteen. Tämä toki toi itselleni hieman lisätyötä, mutta selkeytti samalla kokonaisuuden ja yhden oppitunnin rakennetta eli toimi itselleni samalla työkaluna. Lopuksi lukio-opiskelijat pohtivat sitä onko koodi väline vai osa taidetta sekä sitä, että kuka on tekijä, jos koodi on avointa ja muokattavissa. Tässä kohtaa avoin koodi voitiin nähdä esimerkiksi tuntiharjoituksen kautta, jossa jokainen lukio-opiskelija sai muokata yhdessä kirjoitettua koodia. Pohdinnat koottiin Padlet-alustalle.

Omien havaintojeni mukaan kokonaisuus lähti hyvin käyntiin, vaikka itseäni jännitti paljon. Erityisen myönteistä oli, että opintojaksolle osallistuneita lukio-opiskelijoita oli yhteensä 28. Tämä toisaalta haastoi itseäni ja epäilin, miten ehdin auttaa ja huomioimaan kaikki. Kokemukseni kuvan käsittelyn opettamisesta keväällä antoi ymmärryksen siitä, että teknologian kanssa työskentely voisi olla osalle haastavaa. Oli kuitenkin ilo huomata, että haasteita oli poikkeuksellisen vähän. Tähän varmasti vaikutti, että lukio-opiskelijat työskentelivät alkuun yhteisen tehtävän parissa minun johdollani. Erityisen iloinen olin siitä, että osa opiskelijoista lähti poikkeuksellisen rohkeasti muokkaamaan johdollani kirjoitettua koodinpätkään (Kuva 5) niin kuin olin tehtävänantoon kirjannut. Alkuun ja oikeastaan koko tuntikokonaisuuden ajan jännitin omaa osaamistani. Koen itse olevani aivan alkutekijöissä aiheen parissa ja pohdin erityisesti sitä, että osaanko opettaa aiheita ymmärrettävästi, saavatko opiskelijat itsenäistä työtä tehtyä ja miten ratkaisen ongelmatilanteita. Toisaalta ajattelin, että ollessani itsekkin vasta-alkaja pystyn asettumaan paremmin oppijan asemaan ja huomioimaan opetustahdin sekä kääntää termejä ammattisanastosta yleiskielelle.



Kuva 5. Opiskelijan ensimmäisen tunnin tuotos.

Toisella tunnilla kertailtiin värejä sekä tutustuttiin toistoon ja satunnaisuuteen. Tämä(kin) aihe on laaja, jos siihen haluaisi perehtyä. Kävimme siis vain pintaraapaisella aihetta niin, että siitä saisi mielekkäitä esimerkkejä, jotka olisivat mahdollisimman yksinkertaisia. Esimerkkikoodissa yhdistin for()-silmukan ja random()-funktion (Kuva 6). Tässä koodissa lopputuloksena tietokone piirtää pohjalle ympyröitä 7 pystyyn ja 7 vaakaan, joiden koko ja väri ovat satunnaiset (Kuva 7). Random()-funktio palauttaa annetulta väliltä satunnaiselta vaikuttavan luvun. Tietokoneen kontekstissa satunnaisuus on kuitenkin näennäissatunnaisuutta, sillä luvut tuotetaan matemaattisen kaavan avulla. Tämän vuoksi tulos näyttää satunnaiselta, vaikka se perustuu laskennalliseen prosessiin. Kuvassa 8 lukio-opiskelijan toisen tunnin tuotos, jota hän oli esimerkiksi lähtenyt muokkaamaan omanlaiseksi (vrt. Kuva 7).

```

1
v 2  function setup() {
3     createCanvas(400, 400); //pohjan koko
4     background(220);      // pohjan väri
5     noStroke();           // ei ääri viivoja
6  }
7
v 8  function draw(){
9     // luodaan for -loopiin muuttuja y, jonka arvo on 50. Kun y on pienempi kuin 400 lisätään 50.
10    // eli 50, 150, 200, 250, 300 ja 350. Enempää ei voi lisätä, koska silloin y ei ole pienempi
11    // kuin 400 vaan olisi saman suuruinen. Tehdään sama juttu, mutta muuttujalla x.
v 12   for (let y = 50; y < 400; y += 50) {
v 13     for (let x = 50; x < 400; x += 50) {
14       let r = random(255); // satunnainen punainen arvo
15       let g = random(255); // satunnainen vihreä arvo
16       let b = random(255); // satunnainen sininen arvo
17
18       // täytetään muoto satunnaisilla RGB väreillä
19       fill(r, g, b);
20       circle(x, y, random(10, 50)); // korvataan x ja y luku muuttujilla, jotka luotiin, satunnainen koko
21       noLoop() //draw() ajetaan vain kerran
22     }
23   }
24 }
25 |

```

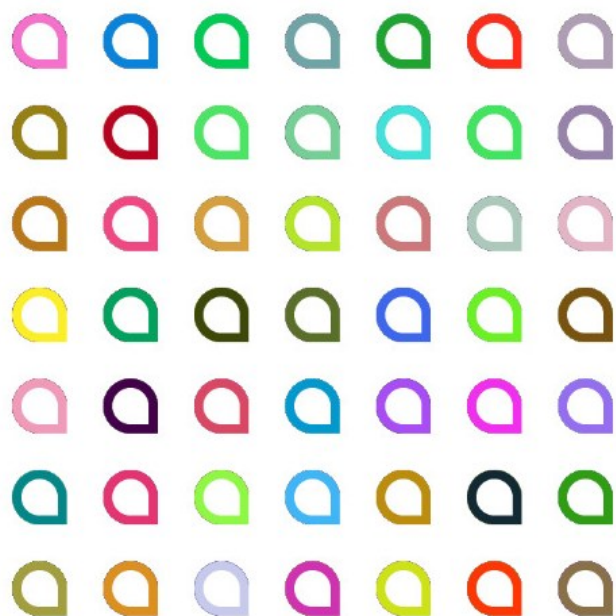
Kuva 6: Esimerkkikoodi toisen tunnin aiheesta



Kuva 7: Toisen tunnin esimerkkikoodin lopputulos

Lopputunnista aika loppui kesken, joten siirsin loppupohdinnan seuraavana päivänä pidettävälle tunnilla.

Toisen tunnin pitäminen oli jo hiukan helpompaa. Toisaalta aihe oli paljon monimutkaisempi kuin edellinen, jossa keskityttiin lähinnä ihan perusasioihin. Pohdin sitä, että onkohan näiden kahden tunnin välillä liian suuri harppaus. Lukio-opiskelijat yllättivät minut jälleen kerran taituruudellaan ja tunnin tehtävä onnistui suhteellisen hyvin. Aika oli tällä tunnilla suuri haaste, joka johtui siitä, että olin laatinut kaksi esimerkkitehtävää yhden sijasta. Jälkeenpäin ajateltuna se oli selvästi liikaa. Tämän tunnin teema olisi ollut järkevintä jakaa kahdelle tunnille tai ainakin jättää toinen esimerkkitehtävä pois.



Kuva 8. Opiskelijan toisen tunnin harjoitustyö.

Kolmannella tunnilla tutustuimme siihen, miten tuoda koodiin interaktiivisuutta. Aiheena oli muuttujat, liike ja ehtolauseet eli jälleen todella laaja paketti, josta pienellä pintaraapaisulla saatiin tuntumaa siihen, mitä niillä tarkoitetaan. Esimerkkikoodissa luotiin muuttuja sekä kokeiltiin käytännössä, miten ehtolause toimii. Lisäksi esimerkkikoodiin tuotiin edelliseltä tunnilta tuttu sattuma. (Kuva 9.) Koodissa muoto seuraa hiiren liikettä ja reagoi näppäimistön painallukseen vaihtamalla muotoa eli tässä tapauksessa valitut muodot olivat ympyrä ja neliö niiden yksinkertaisuuden vuoksi (Kuva 10). Hiirtä klikatessa muodon koko ja väri vaihtuu satunnaisesti.

```

1 let muoto = 'circle'; //luodaan muuttuja, jonka arvoksi annetaan circle
2 let väri; // luodaan muuttuja, jolle ei vielä anneta arvoa
3 let koko = 50; // luodaan muuttuja, jolle annetaan arvoksi 50
4
5 function setup(){
6   createCanvas(400, 400); //pohjan koko
7   väri = color(100, 150, 255); // alkuväri
8   frameRate(24); // ruutua sekunnissa
9   noStroke(); // ei ääri viivoja
10  rectMode(CENTER) // neliö kursorin keskellä
11
12 }
13
14 //tehdään napille funktio. Kun nappia painetaan, jos muoto on ympyrä niin muoto vaihtuu neliöksi
15 // jos muoto on neliö, se vaihtuu ympyräksi.
16 function keyPressed() {
17   if (muoto === 'circle') {
18     muoto = 'rect';
19   } else {
20     muoto = 'circle';
21   }
22 }
23
24 function draw() {
25   background(220); //pohjaväri tähän, koska halutaan, että pohjalla näkyy vain yksi muoto kerrallaan
26
27   // jos hiirtä painetaan annetaan värille satunnainen RGB väri ja muodolle satunnainen koko
28   if (mouseIsPressed) {
29     väri = color(random(255), random(255), random(255));
30     koko = random(20, 100);
31   }
32
33   fill(väri); //täytetään muoto värillä
34
35   // jos muoto on circle, piirretään ympyrä hiiren kursorin kohtaan, jos ei niin piirretään neliö
36   if (muoto === 'circle') {
37     circle(mouseX, mouseY, koko);
38   } else {
39     rect(mouseX, mouseY, koko);
40   }
41 }

```

Kuva 9: Kolmannen tunnin esimerkkikoodi



Kuva 10: Muodot, jotka esiintyvät kolmannen tunnin esimerkkikoodissa

Lopuksi pohdimme sitä, että onko käyttäjä aina se, joka päättää ja millä tavoin esimerkiksi pelit, some ja sovellukset ohjaavat käyttäjää toimimaan tietyllä tavalla.

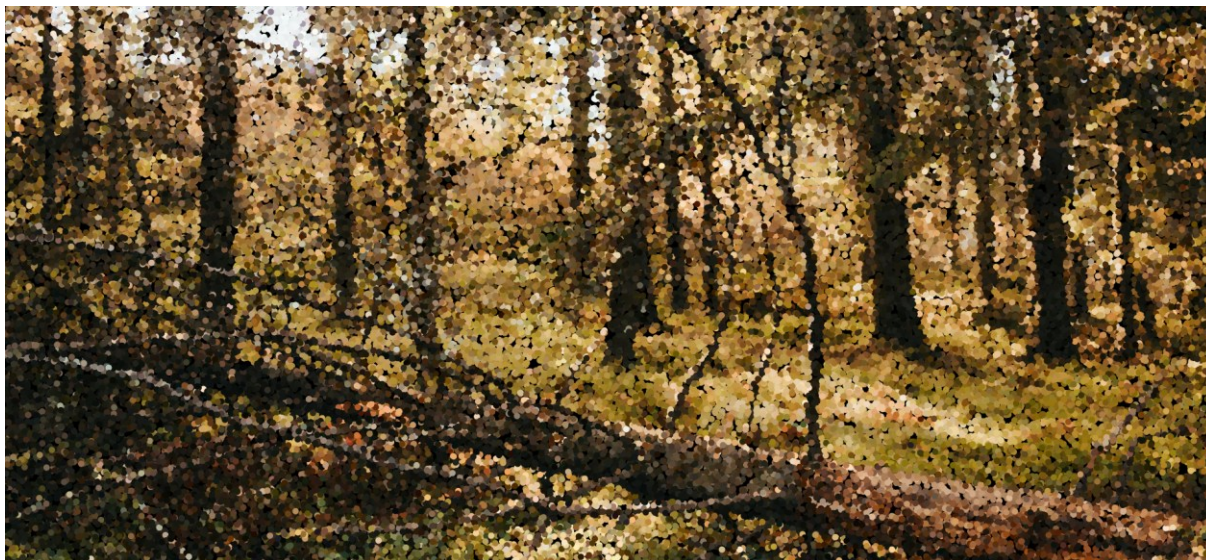
Tällä tunnilla erityisen hedelmällistä oli opiskelijoiden Padlet-keskustelu. Yhteiseen keskusteluun käytin tunnilla enemmän aikaa, kuin muilla, koska jouduin siirtämään edellisen tunnin lop-

pukeskustelun tämän tunnin alkuun. Tunnit olivat onneksi peräkkäisinä päivinä niin oletin asioiden olevan vielä sen verran tuoreessa muistissa, että yhteinen keskustelu olisi vielä mielekäs toteuttaa. Tälläkin tunnilla aika tuntui loppuvan kesken ja esimerkkitehtävä oli luultavasti vaikea hahmottaa, koska vain harva lukio-opiskelija oli lähtenyt muokkaamaan koodia. Tämän vuoksi jälkikäteen ajateltuna myös tämän tunnin aiheet olisi ollut hyvä pilkkoa pienempiin osiin tai pyrkiä tekemään vielä yksinkertaisempi esimerkkitehtävä. Toisaalta tunnin aikataulu oli jälleen tiukka ja siihen lisättyä aiheen vaativuus voi olla myös syy sille, että koodin muokkaamista ei juurikaan toteutettu. En voi aivan suoraa johtopäätöstä vetää siitä, onko syy aiheessa, ajassa vai molemmissa.

Neljännellä ja viidennellä tunnilla lukio-opiskelijat pääsivät työstämään omaa työtään. Neljännän tunnin alkuun esittelin vielä Scottie Chih-Chieh Huangin teoksen Dandelion Mirror (2014–2015). Siinä katsoja seisoo peilimäisen näytön edessä ja järjestelmä tunnistaa katsojan läsnäolon ja alkaa generoida voikukan siemenpallon kaltaista rakennetta. Lisäksi esittelin vielä tunnin lopuksi sen, miten valokuvaa voi hyödyntää omassa työssä, koska kiinnostus sitä kohtaan nousi esiin opitunnin aikana. Tässä hyvin yksinkertaisessa, mutta mielestäni kiinnostavan lopputuloksen tuottamassa esimerkissä näytin, miten valokuva tuodaan osaksi koodia ja mitä sille voi esimerkiksi tehdä (Kuva 11). Tässä esimerkissä tietokone ”piirtää” jatkuvasti valokuvan uudelleen niin, että kuva näyttää eläväiseltä. Kuvaan piirtyy pisteitä, jotka ottavat värin kuvasta siitä kohtaa mihin piirtyvät. Pisteitä suurentamalla kuva piirtyy pehmeästi tuoden melkein pointillisen vaikutelman (Kuva 12).

```
1 |let img;
2
3 |function preload() {
4 |    img = loadImage('metsa.jpg');
5 |}
6
7 |function setup() {
8 |    createCanvas(windowWidth, windowHeight);
9 |    background(0);
10 |    img.resize(windowWidth, windowHeight)
11 |}
12
13 |function draw() {
14 |    for (let i = 0; i < 100; i++) {
15 |        let x = random(width);
16 |        let y = random(height);
17 |        let c = img.get(x, y);
18 |        strokeWeight(10);
19 |        stroke(c);
20 |        point(x, y);
21 |    }
22 |}
23
```

Kuva 11: Esimerkki valokuvan hyödyntämisestä



Kuva 12: Valokuvaesimerkin lopputulos

Kaiken kaikkiaan olin todella ylpeä opiskelijoista. Heidän työstämät itsenäiset teokset onnistuivat yli odotusten. En oikein ollut varma millaisia tuotoksia odottaisi, mutta olin todella iloinen siitä, että kaikki ponnistelivat ahkerasti ja saivat onnistuneen lopputuloksen aikaan. Huomasin myös, että tekemäni valokuvaesimerkki toimi hyvin ja osa hyödynsikin sitä omissa teoksissaan erilaisin varioiden ja omia kuvia käyttäen. Jälkikäteen ajateltuna tämän tyyppisiä esimerkkejä, jossa voidaan kuvaan luoda erilaisia efektejä olisi voinut olla enemmän. Toisaalta se vaatisi myös enemmän aikaa tai jonkin teeman supistamista tai kytkemistä esimerkkiin, jossa valokuvaa tai muuta kuvaa hyödynnettäisiin. Opiskelijat antoivat hyvin palautetta ja osa koki, että tuntikokonaisuus oli pitkä. Todellisuudessa tehtävään käytetty tuntimäärä ei ollut merkittävästi pidempi kuin esimerkiksi edelliseen tehtävään käytetty aika samalla opintojaksolla. Voi olla, että jos tehtävä tuntuu vaikealta ja aihe yleisesti sellainen, mihin ei kuvataiteen opintojaksoilla ole aikaisemmin törmännyt voi aiheuttaa sen, että käytetty aika tuntuu pitkältä.

Opiskelijoiden turhautumista olisi voinut tukea enemmän. Jälkikäteen ajateltuna olisin voinut verrata ohjelmointia minkä tahansa uuden taidon opetteluun.

Koodaamisen raivostuttavin puoli on se, että ei saa mitään aikaiseksi, jos ei mitään osaa. Kynällä ja paperilla ei ole muuta rajoitetta kuin oma mieli. -Opiskelija 5

Moni koki asian vaikeaksi, turhauttavaksi ja rajoittavaksi. Oman mielikuvituksen pohjalta piirtäminen voi tuntua luonnollisesti helpommalta, jos sitä on harjoitellut vuosia, kun taas ohjelmointi voi tuntua vastaavasti rajoittavalta ja vaikealta, jos sitä ei ole aikaisemmin tehnyt. Aivan kuten värien sekoittaminen on joskus tuntunut vaikealta ja mahdottomalta, kunnes taito ja teoreettinen osaaminen on karttunut.

Opintojaksolle sattui opiskelija, joka oli pidempään harrastanut generatiivisen taiteen tekemistä. Tämä oli se tilanne mitä jännitin. Entä jos opintojaksolla on opiskelija, joka tietää ja osaa enemmän kuin minä? Tähän olisi voinut varautua paremmin pohtimalla eriyttämisen keinoja eli minkälaisia haasteita voisın tarjota tällaiselle opiskelijalle. Onnekseni kyseinen opiskelija oli todella itseohjautuva ja haastoi itseään kehittelemällä omia harjoitustöitään liittyen tunnilla käsiteltävään teemaan. Oli ilo huomata, että hän sai jollain tasolla jotain irti ja pystyi haastamaan itse itseään luovin ratkaisuin.

## 4 Opiskelijoiden kokemukset luovasta ohjelmoinnista

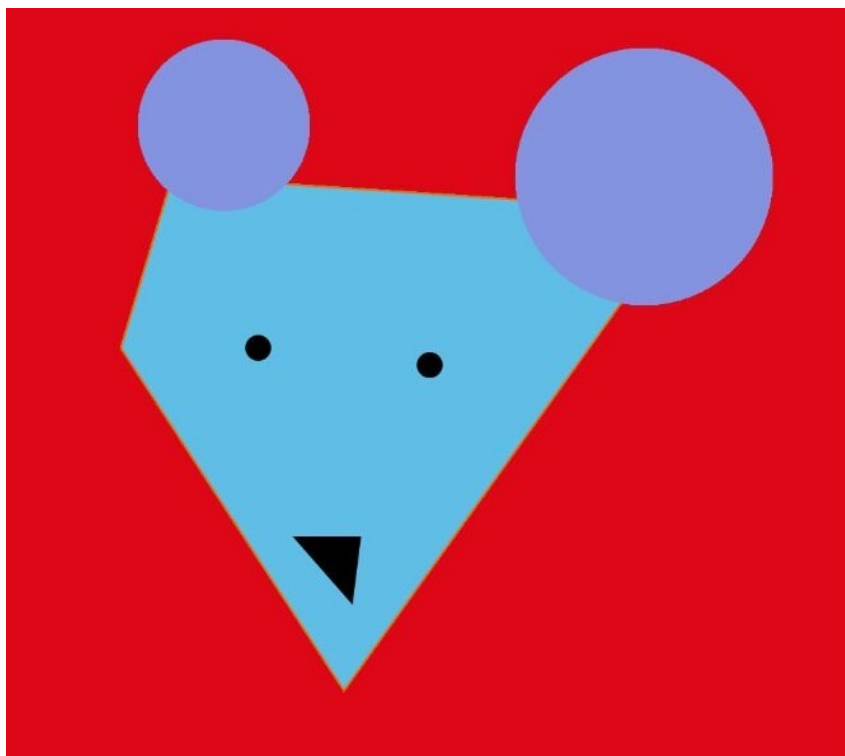
### 4.1 Luova ohjelmointi taiteellisena ilmaisuna

Tässä luvussa tarkastellaan ilmaisua opiskelijoiden koodattujen teosten ja niitä koskevien kirjallisten pohdintojen perusteella. Kun luovaa ohjelmointia tarkastellaan ilmaisukeinona, huomio siirtyy siihen *mitä* lukio-opiskelijat haluavat sanoa ja *miten* he sen tekevät koodin kautta. Tällöin koodattu maailma näyttäytyy paitsi ilmaisun välineenä myös kielenä ja toimintaympäristönä, jossa merkitykset rakentuvat. Ilmaisua ei synny irrallaan teknologiasta vaan rakentuu teknisten prosessien, metaforien, esteettisten valintojen ja tekijyyttä koskevan reflektion kautta. Koodi ei näin ollen näyttäydy ainoastaan toiminnallisena työkaluna vaan ilmaisun rakenteena, jonka avulla lukio-opiskelijat tekevät näkyväksi kokemuksiaan, ajatuksiaan ja tulkintojaan digitaalisesta maailmasta.

Aineiston perusteella ilmaisu näyttäytyy monimuotoisena ja kerroksellisena ilmiönä, sillä lukio-opiskelijoiden koodatut teokset eivät ajaudu pelkkään visuaaliseen estetiikkaan vaan ne sisältävät symbolisia, kokemuksellisia ja käsitteellisiä ulottuvuuksia (kts. sivu 41–43). Dufva (2018) korostaa, että teknologia ei ole neutraali väline vaan kulttuurinen toimija, joka muokkaa tapaa ajatella, nähdä ja toimia. Kun lukio-opiskelijat rinnastavat esimerkiksi värejä vaihtavan planeetan algoritmeihin tai pohtivat tekoälyn roolia taiteessa, he osallistuvat juuri tähän keskusteluun ihmisen ja teknologian välisestä suhteesta. Teoksissa koodi toimii ilmaisun välineenä, jonka kautta lukio-opiskelijat tutkivat sekä omaa luovuuttaan, että suhdettaan teknologiaan ja digitaaliseen kulttuuriin. Ilmaisua ei asetu koodin vastakohtaksi vaan rakentuu sen kautta ja sen rajoja mukailen.

Aineiston perusteella luovuus ei rajaudu opiskelijoiden valmiisiin teoksiin vaan se rakentuu olennaisesti myös yhteisissä harjoituksissa ja työskentelyprosessin aikana (kts. luku 3.2). Opetuskonaisuuden harjoituksissa opiskelijoiden työskentely eteni vaiheittain teknisen perusymmärryksen omaksumisesta kohti kokeilevaa ja leikkimielistä koodin muokkaamista. Alkuvaiheessa tehtävien tavoitteena oli havainnollistaa, miten kone tulkitsee koodia sekä miten visuaalisia elementtejä, kuten muotoja ja värejä ohjataan numeeristen arvojen avulla.

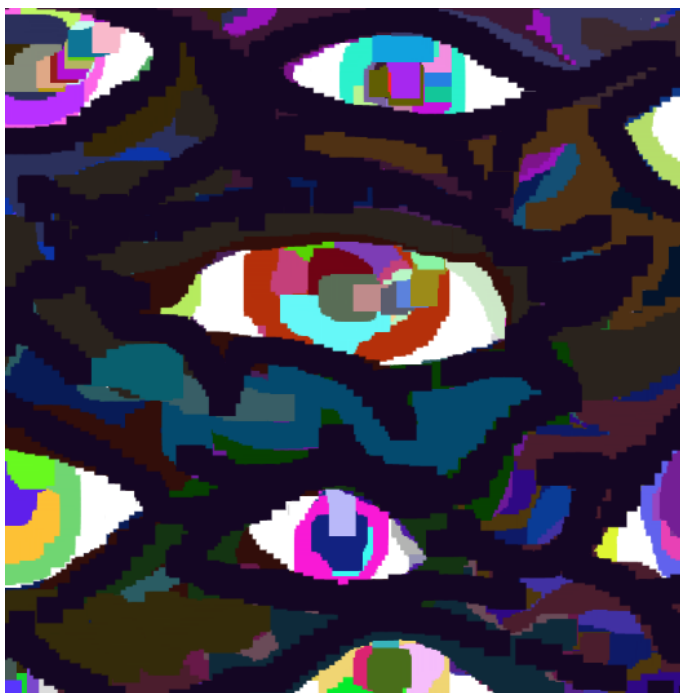
Siirtymä teknisestä ymmärtämisestä kokeilevaan työskentelyyn näkyy aineistossa siinä, että vaikka osa tuotoksista noudatti annettua esimerkkejä, osa opiskelijoista alkoi jo varhaisessa vaiheessa muokata koodia rohkeasti itsenäisesti ja soveltaa sitä esittävien ja abstraktien kuvien rakentamiseen (Kuva 13). Huomioitavaa on se, että tuntitehtävien tehtävänannossa pyysin opiskelijoita muokkaamaan koodia ja kokeilemaan mitä tapahtuu, kun arvoja muutetaan. Osa lähti luomaan esittävää kuvaa (Kuva 13), kun taas osa ei välttämättä muokannut koodia ollenkaan ohjeistuksesta huolimatta tai muokkasi sitä vähän, esimerkiksi vaihtamalla taustan värin. Tämä viittaa siihen, että luova toiminta ei synny vasta teknisen osaamisen ”jälkeen”, vaan kietoutuu osaksi sitä jo oppimisprosessin alkuvaiheessa. Tekninen ymmärrys koodista ja sen toiminnasta ei näin ollen toimi pelkästään luovuuden edellytyksenä vaan myös sen välineenä ja alustana, jonka puitteissa kokeilu ja merkitysten rakentaminen tapahtuvat.



Kuva 13: Opiskelijan ensimmäisen tunnin harjoitustehtävä

Reilu viidesosa (22,73 %) lukio-opiskelijoiden vastauksista painottaa ilmaisua visuaalisen esteetiikan ja tunnelman kautta, jossa kuvan rakentamisen peruselementit nousevat pääilmaisukeinoksi. Teoksen tavoitteena ei välttämättä ole selkeä tai yksiselitteinen sanoma, vaan pikemminkin miellyttävä, kiinnostava tai harmoninen kokonaisuus. Värit, muodot, symmetria ja rytmi nousevat keskeisiksi ilmaisukeinoiksi. Lukio-opiskelijat kuvaavat esimerkiksi halunneensa luoda ”kauniin”, ”harmonisen” tai ”rauhallisen” teoksen, jossa visuaalinen kokonaisuus itsessään kantaa merkitystä. Ilmaisuu rakentuu tällöin katsojan kokemuksen varaan. Teos ei selitä itseään vaan jättää tilaa tulkinalle. Tämä vastaa formalistista taidekäsitystä, jossa esteettinen kokemus voi olla itsessään merkityksellinen. Tässä yhteydessä luova ohjelmointi näyttäytyy ilmaisun muotona, jossa merkitys ei ole ennalta määritelty vaan syntyy avoimessa vuorovaikutuksessa teoksen, tekijän ja kokijan välillä.

Taiteellisessa työskentelyssä symbolinen ilmaisu mahdollistaa sen, että henkilökohtaiset kokemukset, havainnot ja ajatukset voidaan muuttaa jaettaviksi ja yhteisesti tulkittaviksi muodoiksi. Symbolien kautta yksityinen ajattelu saa julkisen muodon. Räsänen (2011, 135) mukaan taiteellinen käsitteellistäminen tapahtuu mielensisäisenä prosessina, jossa kokemuksia jäsenetään visuaalisten ja esteettisten peruskäsitteiden kautta, kuten värin rytmin ja jännitteen kautta. Opiskelijoiden vastauksissa visuaaliset elementit, kuten pallot, silmät tai luonnon muodot eivät toimi ainoastaan kuvan visuaalisina elementteinä vaan rakentavat merkityksiä, joiden kautta lukio-opiskelijat jäsentävät omia kokemuksiaan teknologiasta, koodaamisesta ja digitaalisesta kulttuurista.



Kuva 14. Opiskelijan Koodattu maailma -teos.

Symbolit eivät ole yksiselitteisiä vaan ne avaavat tilaa henkilökohtaiselle tulkinnalle. Taiteessa merkitys ei välity suoraviivaisesti vaan ne kutsuvat katsojan osallistumaan merkityksen rakentamiseen. Opiskelijoiden koodatuissa teoksissa symbolinen ilmaisu toimii keinona käsitteellistää ilmiöitä, joita on vaikea sanallistaa, kuten post-digitaalisen maailman läsnäoloa, kontrollia, sattumaa tai ihmisen ja teknologian välistä suhdetta. Taiteellinen ilmaisu ei rajaudu valmiiseen lopputulokseen vaan se rakentuu prosessin aikana syntyvistä oivalluksista. Teos ja sen tekeminen toimivat ajattelun välineenä, jonka avulla lukio-opiskelijat kokevat ja tutkivat maailmaa sekä omaa asemaansa sen osana. Symbolinen ilmaisu mahdollistaa etäisyyden ottamisen omaan tekemiseen ja tarjoaa kehyksen, jossa henkilökohtaiset kokemukset voidaan liittää laajempiin kulttuuriin ja filosofisiin kysymyksiin. Näin taiteellinen ilmaisu toimii sekä ajattelun, että viestinnän välineenä ja koodi sen mahdollistajana. Opiskelijoiden teoksissa esimerkiksi pallot symboloivat tietoa, algoritmeja ja informaatiota, kun taas silmät viittaavat valvontaan, näkyvyyteen tai sosiaaliseen mediaan (Kuva 14). Luontoon viittaavat elementit puolestaan asettuvat vastakkain digitaalisen ympäristön kanssa tai toimivat sen jäljitteinä, jolloin ne avaavat pohdintaa ihmisen ja teknologian suhteesta.

Teema liittyy koodattuun maailmaan siten, että kuvassa liikkuu sinisiä epämääräisiä palloja, jotka kuvastaa teoksessa liikkuvaa tietoa, kun tekee koodia. Pallojen liikkuminen on epämääräistä, koska näin voisin kuvitella joidenkin tietojen liikkuvan koneissa. Valkoiset pallot ovat ns. pölyä, jota on kertynyt laitteiden rakenteisiin. -Opiskelija 11

Näiden symbolisten ratkaisujen kautta ilmaisu laajenee yksittäisestä visuaalisesta elementistä kohti kokemuksellista ja filosofista pohdintaa. Koodatun teoksen tekeminen näyttäytyy lukio-opiskelijalle keinona reflektoida esimerkiksi kontrollia ja sattumaa sekä yksilön vaikutusmahdollisuuksia. Kokemus siitä, että pieni muutos koodissa voi muuttaa koko lopputuloksen saa symbolisen merkityksen ja rinnastuu arjen ja/tai yhteiskunnan rakenteisiin. Tämä tukee Dufvan (2018, 52–54) ajatusta siitä, että käyttämällä luovaa koodausta pedagogisena menetelmänä opiskelija voi ilmaista itseään ja muokata ymmärrystään ja yhteyttä ympäristöönsä.

Aineistossa käy ilmi pohdinta ihmisen ja koneen välisestä suhteesta. Esimerkiksi lukio-opiskelijat pohtivat missä määrin kone tai tekoäly on luova toimija ja mikä on ihmisen rooli taiteellisessa prosessissa. Tekoäly näyttäytyy opiskelijoille työkaluna, haastajana ja keskustelukumppanina. Ilmaisun kohteena ei ole tällöin ainoastaan visuaalinen lopputulos vaan itse taiteen tekemisen ehdot digitaalisessa kulttuurissa. Teokset kommentoivat luovuuden rajoja, tekijyyttä, vallankäyttöä sekä sitä, missä määrin kone voi olla luova toimija. Näissä pohdinnoissa koodi näyttäytyy tekniistä harjoitusta laajempina ilmiönä. Se on sosiaalinen, poliittinen ja kulttuurinen toimija.

(Knochel & Patton 2015, 28). Teos ei ainoastaan esitä jotain vaan kyseenalaistaa taiteen tekemisen ehtoja digitaalisessa ympäristössä ja lukio-opiskelijan asemaa osana teknologisesti välittyntä todellisuutta.

Haluan työlläni nostaa esiin, kuinka kone voi olla ihmistä nopeampi tekemään taide, jolloin tuntuu että koneet ovat ihmisen edellä taiteen tekemisessä. Koneen luoma taide on kuitenkin yleensä ihmisen ohjailemaa. Haluan työni avulla pohtia sitä, onko koneen tekemä taide yhtä arvokasta kuin ihmisen tekemä, ja onko taide ihmisen tekemää, jos siinä on käytetty apuna konetta ja koodeja. -Opiskelija 4

Aineistossa on myös vastauksia, joissa lukio-opiskelijat eivät tunnista teokselleen tiettyä ilmaisullista tavoitetta tai kokevat vaikeaksi sanallistaa sitä. Teos on saatettu tehdä sen takia, että se oli ”pakko” tehdä ilman tietoista sanomaa. Tämä havainto osoittaa sen, että ilmaisun reflektio ei ole itsestään selvä osa kaikkien opiskelijoiden työskentelyä vaan taito, joka kehittyy kokemuksen ja ohjauksen myötä. Ilmaisun puuttuminen sanallisella tasolla ei välttämättä tarkoita ilmaisun puuttumista teoksessa vaan vaikeutta tunnistaa tai nimetä sitä. Tämä korostaa pedagogisen tuen merkitystä luovan koodaamisen kontekstissa.

## 4.2 Ohjelmointi oppimiskokemuksena

Kokonaisuutena ohjelmointiteema rakentuu aineistossa ristiriitaisena mutta rikkaana kokemuksena. Se on samanaikaisesti vaikeaa ja innostavaa, rajoittavaa ja vapauttavaa. Ohjelmointi pakottaa lukio-opiskelijat kohtaamaan omat taitonsa, kärsivällisyytensä ja suhtautumisensa virheisiin, mutta antaa samalla myös mahdollisuuden uudenlaiseen ilmaisuun ja ajatteluun. Näin ohjelmointi ei toimi pelkästään teknisenä taitona vaan merkittävänä osana taiteellista oppimisprosessia ja digitaalisen kulttuurin ymmärtämistä. Se myös haastaa lukio-opiskelijoiden käsityksiä taiteesta ja taiteellisesta tekemisestä ja laajentaa näin ollen myös ymmärrystä siitä, kuinka laaja taiteen kenttä voi olla.

Lukio-opiskelijat liittyivät ohjelmoinnin muun muassa laajempaan digitaaliseen kulttuuriin, jossa algoritmit, tekoäly ja jatkuva muutos muodostavat kontekstin, jossa teokset syntyvät. Koodi ei ole vain tekninen työkalu vaan osa lukio-opiskelijan ilmaisua ja ajattelua. Teokset toimivat symbolisina ja kokemuksellisina tulkintoina algoritmisesta maailmasta, yksityisyydestä ja interaktiivisuudesta. Tekoäly nousee osaksi keskustelua tekijyydestä, luovuudesta ja digitaalisen ilmaisun ehdoista. Näin ohjelmointi näyttäytyy prosessina, jossa digitaalisen maailman rakenteet ja ihmisen oma toimijuus kohtaavat.

Se (teos) pystyisi ainakin herättämään ajatuksia oman kuplan tuhoutumisesta, jos jopa kyseenalaistaisikin digitaalista maailmanmenoa. -Opiskelija 8

Ohjelmointi hahmottuu aineistossa luovana välineenä, jonka avulla voidaan tuottaa visuaalisesti monipuolisia ja esteettisesti merkityksellisiä teoksia. Lukio-opiskelijat kuvaavat oivallusta siitä, että koodin avulla on mahdollista tehdä taidetta ja ilmaista itseään samalla tavalla kuin perinteisimmillä taiteen menetelmillä. Koodi rinnastetaan siveltimeen tai maalaamiseen, mutta sen luonne koetaan erilaisena - vaikeampana. Luovuus rakentuu loogisten sääntöjen, toiston ja parametrien varaan. Ohjelmointi mahdollistaa generatiivisen ajattelun ja dynaamisten kokonaisuuksien hallinnan. Samalla se ei ole kuitenkaan täysin neutraali väline vaan koodin tekninen loogikka ja syntaksi määrittää sen mitä on mahdollista ilmaista. Tämä rajoite pakottaa neuvottelemaan omien esteettisten tavoitteiden ja koodin tarjoaman mahdollisuuden välillä.

Oivalsin koodatessa, että taidetta on muutkin asiat kuin vaan perus maalaukset, veistokset ja piirrokset. -Opiskelija 14

Ohjelmointi koetaan kuitenkin rajoittavan omia ilmaisukeinoja. Useat lukio-opiskelijat tuovat esiin kokemuksen siitä, että oma idea on vaikea toteuttaa, jos koodauksen periaatteita ei hallitse riittävästi. Tämä johtaa tilanteisiin, joissa omat ideat koetaan hyväksi, mutta tekninen osaaminen ei mahdollista niiden toteuttamista. Ohjelmointi vaatii kykyä visualisoida samanaikaisesti lopputulos, että sitä tuottava koodi mikä tekee luovasta prosessista monitasoisen ja kuormittavan. Vastaavasti ohjelmoinnin ollessa tuttua sitä ei koeta rajoittavana.

-- mielestäni maalaaminen tai käsillä tekeminen on luovempaa ja mukavampaa. -Opiskelija 6

Digitaalisen työskentelyn kokeminen abstraktimpana suhteessa materiaaliseen tekemiseen on linjassa Saethre-McGuirkin (2022, 2; 51) kanssa, joka korostaa materiaalisen työskentelyn moni-aistisuutta ja vuorovaikutusta keskeisenä osana taiteellista prosessia. Digitaalisessa ympäristössä tämä suhde muuttuu, mikä vaikuttaa myös tekijän kokemukseen tekemisestä.

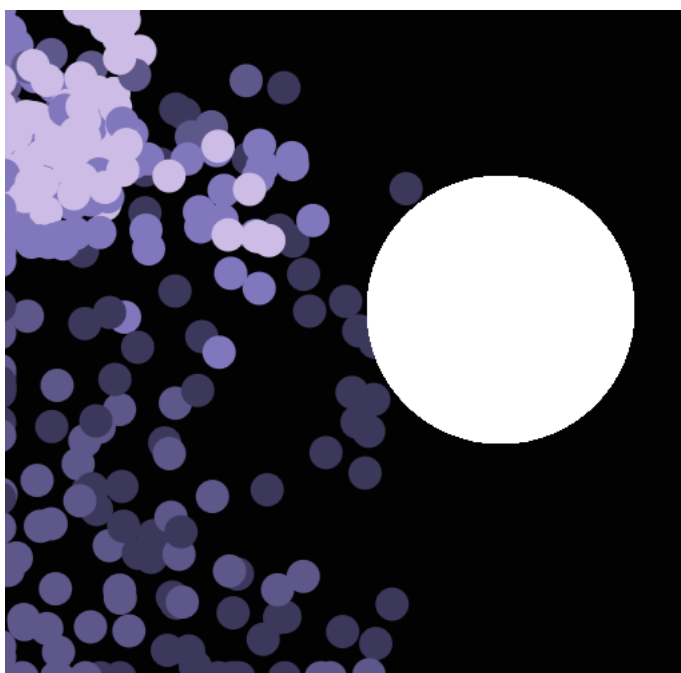
Ohjelmoinnin oppiminen näyttäytyy aineistossa vahvasti kokemuksellisenä. Monet lukio-opiskelijat kuvaavat koodaamisen aluksi vaikeaksi, jopa ahdistavaksi, mutta samalla yllättäväksi ja pal-kitsevaksi. Tämä ristiriitaisuus tuottaa oivalluksia sekä itse välineestä/materiaalista, että omasta toimijuudesta. Aineistossa mainittiin yllätyksestä siitä, että koodin avulla voi tuottaa visuaalisesti vaikuttavia ja merkityksellisiä teoksia ja että ohjelmointi voi toimia taiteellisenä ilmaisukeinona muiden taiteenlajien rinnalla.

Koodin tekeminen taiteessa on kiehtovaa, koska koodi yleensä yhdistetään olennais-  
ten sovelluksien luomiseen. Tarkoitan siis sitä, että koodi on työkalu avustamaan  
ihmisiä saamaan aikaiseksi asioita. Konsepti siitä, että koodaamista voi käyttää  
myös leikkimiseen on yllättävää ja hieno asia. -Opiskelija 5

Knochel ja Patton (2015, 25) keskustelevat siitä, miten arvokasta on, että taideopiskelijat harjoit-  
tavat ohjelmoinnista ajattelua luomalla leikkimielisesti omaa koodia tietokoneella, joka on sekä  
taiteen väline että tilaisuus pohtia kriittisesti sitä, miten ohjelmoitava digitaalinen media muok-  
kaa yhteiskuntaa. Ohjelmointi ja siihen lisätty taiteellinen ilmaisu tuo leikkikentän luovalle työ-  
kentelylle. Pohdinnoissa lukio-opiskelijat kokivat oppivansa koodin rakennetta ja sen luomista,  
mutta nostavat samalla esiin myös kriittistä pohdintaa liittyen post-digitaalisen maailman ilmiöi-  
hin.

Opiskelijoiden pohdinnoissa ohjelmointi hahmottuu prosessina, joka on samanaikaisesti vaativa  
ja oivalluksia synnyttävä, mutta myös tunteita herättävä. Koodaaminen ei näyttäydy vain tekni-  
senä suorituksena vaan oppimis- ja ajatteluprosessina, jossa pienet yksityiskohdat voivat vaikut-  
taa ratkaisevasti lopputulokseen. Suurimmalle osalle lukio-opiskelijoista koodaaminen on uusi  
työskentelytapa. Tämän vuoksi pohdinnoissa korostuu epävarmuuden ja keskeneräisyyden ko-  
kemukset. Lisäksi lukio-opiskelijat tekivät havaintoja siitä, kuinka pienikin virhe tai muutos koo-  
dissa voi johtaa koko teoksen epäonnistumiseen, mikä tekee ohjelmoinnista tarkkaa ja vaativaa.  
Se loi turhautumisen tunteita, mutta myös oivalluksia oppimisen prosessista.

Oli minulle yllätys, että koodaamiseen ei välttämättä tarvitse osata liikaa matema-  
tiikkaa. Jos osaa jonkin verran englantia, se on jo hyvä pohja. Koodaaminen ei siis ole  
mitään rakettitiedettä! -Opiskelija 8



Kuva 15. Opiskelijan ”Koodattu maailma”-tehtävän työ.

Kontrollia voitiin käsitellä teoksen aiheena (Kuva 15), mutta myös ohjelmointiin liitettiin kokemus, joka korostaa kontrollia. Teoksen idea saattoi olla selkeä, mutta koodin logiikka ja sen tuomat ennakoimattomat seuraukset loivat haastetta itse toteuttamisessa. Tämä ilmenee erityisesti kuvauksissa vahingoista ja virheistä, jotka joko koetaan turhauttavina tai päätetään säilyttää osana teosta. Virhe ei tällöin ole pelkästään epäonnistuminen vaan se voi toimia kipinästä luovalle prosessilla ja avata uusia mahdollisuuksia. Tämä asettaa ohjelmoinnin lähelle kokeellista taiteellista työskentelyä, jossa kontrolli ja ennakoimattomuus ovat jatkuvassa yhteydessä.

Supersankarin hiukset eivät olleet ihan miten ne halusin, mutta koska ne näyttivät niin hauskoilta ja epätodennäköisiltä päätin pitää ne. -Opiskelija 17

Ohjelmointi rakentaa suhdetta aikaan ja kärsivällisyyteen. Vastauksissa toistuu kokemus siitä, että rajallinen aika ja vähäinen ennakkotaito rajoittavat mahdollisuuksia toteuttaa omia ideoita. Samalla tämä rajoite pakottaa yksinkertaistamaan, tekemään valintoja ja hyväksymään keskenolun. Ohjelmointi näyttäytyy tällöin kompromissina ideaalisen vision ja käytännön toteuttavuuden välillä.

Lisäisin katsojan mahdollisuutta muuttaa asioita työssä tai muuntaisin värien vaihtumisen komentoa erilaiseksi. -Opiskelija 3

Ohjelmointiin liittyvät oppimisen ja hallinnan teemat korostuvat, kun lukio-opiskelijat pohtivat teostensa mahdollista uudelleentyöstä. Vastauksissa mainitaan tarve harjoitella enemmän, suunnitella ideaa huolellisemmin tai valita tekniikka, joka vastaisi paremmin omaa ajatusta. Samalla osa opiskelijoista ilmaisee tyytymättömyyttä lopputulokseen juuri sellaisenaan erityisesti suhteessa omaan lähtötasoonsa. Tämä viittaa ohjelmoinnin rooliin minäpystyvyyden rakentajana. Vaikka taidot koetaan rajallisiksi, onnistuminen koetaan silti merkitykselliseksi.

Lukio-opiskelijat ymmärtävät koodia ja digitaalista kulttuuria tekemisen, kokeilun ja ilmaisun kautta, vaikka he eivät vielä hallitse täysin ohjelmoinnin käsitteitä tai teknistä logiikkaa. Digi-grasping tarkoittaa tapaa ymmärtää digitaalisuutta kokemuksellisesti ja kehollisesti ilman, että ymmärrys perustuu ensisijaisesti käsitteisiin, teorioihin tai tekniseen tietoon. Kyse ei siis ole siitä, että ihminen osaisi selittää, miten digitaaliset järjestelmät toimivat vaan siitä, että hän aistii, tunnistaa ja hahmottaa digitaalisuuden osana omaa elämäänsä ja ympäristöään. (Dufva & Dufva 2019).

Työni ideana on, että katsoja voi itse koneen hiirellä piirtää teokseen. Katsoja voi yrittää kilpailla koneen kanssa, kumpi pystyy täyttämään teoksen mustan taustan ensimmäisenä. -Opiskelija 4

Lukio-opiskelijan kuvaus omasta teoksestaan konkretisoi digi-graspingin periaatetta, jossa digitaalinen järjestelmä ymmärretään ensisijaisesti kokemuksen kautta. Katsojan ja koneen välinen ”kilpailu” tuo näkyväksi eron inhimillisen ja digitaalisen toiminnan välillä ilman, että järjestelmän teknistä logiikkaa tarvitsee selittää. Hiiren liike, ruudulla tapahtuva jälki ja koneen automaattinen vastaus muodostavat kokemuksellisen kokonaisuuden, jossa ihminen ja kone asettuvat samaan tilaan toimijoina. Lukio-opiskelija ei selitä koodin rakennetta, mutta osoittaa ymmärtävänsä sen logiikkaa kokemuksellisesti. Tämä vastaa digi-graspingin ajatusta siitä, että digitaalista maailmaa voidaan hahmottaa aistimalla ja toimimalla sen kanssa ilman, että ymmärrys edellyttää teknisten käsitteiden hallintaa.

Digitaalisuus tulee tässä teoksessa koetuksi vuorovaikutuksena fyysisessä toiminnassa. Koneen läsnäolo tuntuu samanaikaisesti tutulta ja etäiseltä. Taiteellinen prosessi toimii välineenä, joka tekee digitaalisuuden aistittavaksi ja hahmotettavaksi ilman käsitteellistä selittämistä. (Dufva & Dufva 2019). Luova ohjelmointi mahdollistaa digitaalisten prosessien tutkimisen kehollisesti ja kokemuksellisesti. Teos ei ainoastaan esitä digitaalisuutta vaan asettaa katsojan tilanteeseen, jossa digitaalinen järjestelmä tulee koetuksi toiminnan, vuorovaikutuksen ja ajan kautta. Myös Saethre-McGuirkin (2022, 41) jakaa näkemyksen siitä, että kehollinen kokemus ei katoa digitaalisessa ympäristössä, vaan tekijä tuo mukanaan kehollisen tietonsa ja toimintatapansa myös digitaalisen tilaan.

Ohjelmoinnin avulla opiskelijat pohtivat digitaalista maailmaa. Dufvan & Dufvan (2019) koodimetaforissa esiintyy kulttuurimetafora, joka asettaa koodin ymmärtämisen välineeksi, kuten esimerkiksi eräs lukio-opiskelija kuvaa kertoessaan teoksensa ideaa. Kuvaus vastaa pitkälti Dufva & Dufvan (2019, 102–103) kuvailua koodin kulttuurimetaforasta.

Filosofinen idea teokseeni oli seuraavanlainen: hyvin hyvin suuri määrä arjessa käyttämiämme asioita sisältää koodia tai on tehty koodin avulla, mm. kaikki tekniset laitteet. Suurin osa tästä koodista on riippuvainen vapaaehtoisten aloittamista ja pyörittämistä avoimen lähdekoodin projekteista, joiden koko kehityshistoriaa voi lukea läpi nettiarkistoista sekä seurata livenä tulevaisuuden kehitystä. Kehitykseen voi jopa vaikuttaa esittämällä omia argumentteja projektien keskustelupalstoilla. Tämä yhteisöllinen kehitys tiivistyy sosiaalisilla koodialustoilla, kuten GitHub, GitLab, ForgeJo, Tangled..., joidenka sisältämä data muodostaa lopulta pohjan "koodatulle maailmalle". -Opiskelija 1

Ohjelmointi ei siis ole pelkästään tekninen työkalu vaan se voidaan ymmärtää myös materiaalina luovalle ilmaisulle.



Kuva 16. Xyny – Tangletree 2025

### 4.3 Luova ohjelmointi merkityksellisenä kokemuksena

Tässä teemassa aineisto siirtyy yksittäisistä teoksista ja teknisistä ratkaisuista kohti opiskelijoiden itseymmärrystä ja käsityksiä taiteesta, teknologiasta ja luovuudesta. Lukio-opiskelijoiden vastaukset osoittavat, että koodattu taide toimii paitsi ilmaisun ja teknisen oppimisen välineenä, myös tilana, jossa oma suhde luovuuteen, osaamiseen ja teknologiaan tulee näkyväksi. Tämä teema syventää ymmärrystä siitä, miten taiteellisen työskentely digitaalisessa ympäristössä vaikuttaa opiskelijoiden käsitykseen itsestään ja taiteen roolista nykyculttuurissa.

Opiskelijoiden kokemukset ylpeydestä ovat aineistossa moniäänisiä. Osa kokee ylpeyttä konkreettisesta lopputuloksesta, kuten värien harmoniasta, liikkeestä tai visuaalisesta yksinkertaisuudesta. Toiset taas liittävät ylpeyden ensisijaisesti prosessiin eikä niinkään teoksen esteettiseen tasoon. Ylpeyden aiheena on myös se, että työ ylipäättään valmistui tai että tekijä jaksoi jatkaa turhautumisesta huolimatta. Luovan ohjelmoinnin kokemus voi muuttaa tai ainakin luoda positiivista vahvistusta oppijalle itselleen omasta toimijuudesta ja minäpystyvyydestä.

Olen ylpeä että osasin tehdä jotain mitä en ollut koskaan kokeillut. -Opiskelija 6

Ylpeyden vastaparina aineistossa esiintyy myös epävarmuutta ja vähättelyä. Osa opiskelijoista ilmaisee, että ei ole erityisen ylpeitä mistään tai että työ on tehty mallin mukaan. Tämä asettuu vastakkain niiden vastausten kanssa, joissa jo pienet onnistumiset esimerkiksi yksinkertaisen, mutta toimivan ratkaisun saavuttamiseen koetaan merkitykselliseksi. Aineisto osoittaa, että ohjelmointiin liittyvä vieraus vaikuttaa voimakkaasti siihen, miten lukio-opiskelijat arvioivat omaa osaamistaan ja onnistumistaan.

Lukio-opiskelijat kuvaavat yllättyneensä erityisesti siitä, kuinka taiteellista luova ohjelmointi voi olla ja kuinka monipuolisia teoksia sen avulla on mahdollista luoda. Useille ajatus siitä, että koodaaminen ei ole vain tekninen tai matemaattinen taito vaan väline taiteelliseen työskentelyyn oli uutta. Eli voidaan ajatella, että luova ohjelmointi laajentaa lukio-opiskelijan taidekäsitystä. Tässä tapauksessa voidaan puhua transformatiivisesta oppimisesta jossa oppijan ennakkokäsityksenä ohjelmointi ei liity taiteeseen, mutta tuntikokonaisuuden jälkeen havainto on muuttunut ja ohjelmointi liitetään osaksi taidetta (Anttila 2022, 1:3).

En koskaan tullut ajatelleeksi, että koodaamisella voisi oikeasti tehdä taidetta, ja että sitä varten on kehitetty alustoja ja avustavia nettisivuja. Yllätyin positiivisesti siitä, että tämä on oikea vaihtoehto taiteelle. -Opiskelija 5

Lukio-opiskelijat pohtivat sitä, miten kokevat itsenä luovana tekijänä. Osa vastauksista on kriittisiä ja jopa kielteisiä. Jotkut kokevat, etteivät ole luovia tai etteivät pidä koodaamisesta lainkaan. Näissä vastauksissa korostuu kokemus rajoitetusta ilmaisun vapaudesta ja turhautuminen silloin, kun oman idean toteutukseen ei ole itsellä tarpeeksi osaamista tai aikaa. Toisaalta monet lukio-opiskelijat tunnistavat itsessään uusia tai vahvistuneita piirteitä, kuten kyvyn ideoida ja kyvyn visuaaliselle ajattelulle ja näkemiselle.

Että jos vain haluat onnistua jossain niin se kyllä onnistuu kun vain haluaa nähdä vai-  
vaa. -Opiskelija 13

Yksi opiskelijoista nimeää suoraan järjestelmällisen tai loogisen luovuuden omaksi vahvuudekseen. Tämä viittaa siihen, että koodaaminen avaa uudenlaisen tavan hahmottaa omaa luovaa identiteettiä, joka ei perustu spontaaniin ilmaisuun vaan rakenteiden ja sääntöjen hyödyntämiseen. Luovalla ohjelmoinnilla voidaan siis antaa uusia välineitä lukio-opiskelijan taiteelliselle ilmaisulle. Tämä voi tuoda taiteen pariin sellaisia, jotka eivät koe omakseen perinteisiä menetelmiä vaan ovat kiinnostuneita enemmän teknologiasta ja digitaalisesta ympäristöstä. Siinä missä Saethre-McGuirkin (2022) painottaa digitaalisen työskentelyn rajoitteita ja materiaalisen vuoro-vaikutuksen puutetta, tämä esimerkki kertoo sen, että koodi voidaan kyllä kokea rajoittavana, mutta toisaalta se voi toimia myös uusien ideoiden ja ilmaisumuotojen tuottajana.

Taide ei näyttäyty vain esteettisenä toimintana vaan myös merkityksellisenä ajattelun ja viestinnän muotona. Useat lukio-opiskelijat mainitsevat, että prosessi sai heidät ajattelemaan taidetta syvällisemmin ja pohtimaan teoksen sanomaa. Tämäkin esimerkki heijastaa transformatiivisen oppimisen tapahtumaa, jossa oppijan aikaisempi käsitys laajenee tai muuttuu (Anttila 2022, 1:3).

Opin ajattelemaan taidetta syvällisemmin sekä niin, että taideteoksillakin on jokin sanoma. -Opiskelija 10

Jossain vastauksissa nousee esiin kriittinen suhde tekoälyyn ja algoritmeihin, toisaalta niiden mahdollisuudet ja toisaalta uhka taiteen korvautumisesta. Tämä heijastaa laajempaa kulttuurista keskustelua, jossa teknologia nähdään samanaikaisesti välineenä, inspiraation lähteenä ja haasteena luovuudelle.

Opiskelijat olivat asettaneet itselleen omat tavoitteet tuntikokonaisuuteen liittyen. Suurin osa koki saavuttaneensa asettamansa tavoitteet, mutta kaksi vastaajaa koki, ettei saavuttanut itselleen asettamia tavoitteita, koska aikaa oli liian vähän työn valmiiksi saamiseen tai koki, että koodin ymmärtäminen on niin haastavaa, että motivaatio loppui kesken, vaikka olisi halunnut kehittyä siinä. Tämä palaute tukee myös omia havaintojani tuntikokonaisuuden kiireestä ja ehkä liian laajasta kokonaisuudesta aikaan nähden.

## 5 Johtopäätökset

Tämän tutkimuksen lähtökohtana oli tarkastella, miten luovan ohjelmoinnin opetuskokonaisuus voidaan toteuttaa osana lukion kuvataiteen opetusta ja miksi sitä tulisi opettaa osana lukion kuvataiteen opetusta. Tulosten perusteella luova ohjelmointi ei näyttäyty ainoastaan teknisenä taitona, vaan se avaa kuvataiteen opetuksessa tilan, jossa taiteellinen ilmaisu, digitaalisen kulttuurin kriittinen tarkastelu sekä kokemuksellinen oppiminen kietoutuvat toisiinsa.

### 5.1 Ohjelmoinnin opettamisesta

Tässä tutkimuksessa reaaliaikainen koodaus yhdistettiin opetuskokonaisuuteen, jossa lukio-opiskelijat harjoittelivat ensin yhteisen esimerkkikoodien avulla ja toteuttivat oppimansa pohjalta itsenäisen teoksen ”Koodattu maailma” -tehtävänannon mukaisesti. Tulosten perusteella menetelmä tuki sekä teknisten taitojen omaksumasta että luovan ajattelun ja merkityksen rakentamisen prosessia. Voidaan tehdä oletus, että reaaliaikainen koodaus yhdistettynä kokeilevuuteen teki siitä pedagogisesti mielekkään välineen luovan ohjelmoinnin opetuksessa nykyaikapedagogiikan periaatteita mukaillen. Luovan ohjelmoinnin opetuskokonaisuus osoittautui toimivaksi,

kun se rakennettiin rajatuksi osaksi Nykyaide -opintojaksoa ja suunniteltiin korostamaan kokemuksellista ja merkityksellisiä ulottuvuuksia teknisten taitojen sijaan. Luovan ohjelmoinnin kokonaisuuden mielekkyyttä korostaa myös se, että valtaosa opintojaksolla olleista piti sitä positiivisena kokemuksena.

Huomionarvoisena tuloksena nostan tutkimukseni tuloksiin nojaten sen ajatuksen, että luovassa ohjelmoinnissa tekninen ja taiteellinen toiminta eivät ole erillisiä. Se tulee huomioida opetusta suunniteltaessa. Toteuttamalla tällä tavoin luovan ohjelmoinnin kokonaisuuden se mahdollista kokeilevuuden ja matalan kynnyksen osallistumisen. Reaaliaikainen koodaus ja opettajajohtoinen alkuvaihe loivat rakenteen, joka tuki opiskelijoita alkuun pääsemisessä ja mahdollisesti vähensi ohjelmoinnin koettua vaikeutta. Tämä tukee aiempaa tutkimusta (Brown & Wilson 2018; Fuentes Martinez 2024), jossa korostetaan reaaliaikaisen ohjelmoinnin menetelmän merkitystä ohjelmoinnin ajattelun näkyväksi tekemisessä. Samalla nikkarointi (tinkering) vaihe mahdollisti opiskelijoiden oman ilmaisun kehittymisen ja koodin muokkaamisen henkilökohtaiseksi ilmaisun välineeksi.

Tulokset kytkeytyvät myös post-digitaalisen taidekasvatuksen näkökulmaan (Cramer 2014; Saethre-McGuirk 2022; Slotte Dufva 2021), jossa digitaalinen teknologia ei ole erillinen oppimisen alue vaan osa arjen, materiaalisen ja kulttuurisen todellisuuden kokonaisuutta. Opiskelijoiden teemat, kuten tekoäly, algoritmit ja ihmisen ja koneen välinen suhde osoittavat, että luova ohjelmointi mahdollisti digitaalisen kulttuurin ilmiöiden käsittelyn taiteellisen työskentelyn kautta. Näin ollen ohjelmointi ei jäänyt tekniseksi harjoitukseksi. Sen avulla lukio-opiskelijat jäsentivät omaa suhdettaan digitaaliseen ympäristöön taiteellisen työskentelyn kautta, jota yhteinen keskustelu tuki. Tätä voidaan tarkentaa myös Dufvan & Dufvan (2019) kuvaaman digi-graspin-käsitteen avulla. Opiskelijoiden kokemuksissa korostui tekemisen, kokeilemisen ja vuorovaikutuksen kautta rakentuva ymmärrys ohjelmoinnista ja digitaalisuudesta. Oppiminen ei perustunut pelkästään käsitteelliseen ymmärrykseen algoritmeista vaan siihen, että digitaaliset rakenteet tulivat näkyviksi ja kehollisesti koettaviksi tekemisen kautta. Esimerkiksi tehtävä, jossa käsiteltiin sattumaa ja algoritmeja avasi opiskelijoille näkymän siihen, miten digitaaliset järjestelmät tuottavat ”sattumaa” ja ohjaavat valintoja arjessa.

Tuntien rakenne tuki myös ohjelmoinnillisen ajattelun kehittämistä. Tuntitehtävien aikana hyödynnettiin toistorakenteita ja ehtolauseita, jotka liittyvät ohjelmoinnillisen ajattelun opettamiseen. Teokset toteutettiin tekstipohjaisella ohjelmoinnilla. Jotta opiskelija saisi teoksen tehtyä niin hänen tulisi myös osata tulkita tekstipohjaista ohjelmointikieltä eli tässä tapauksessa Ja-

vaScriptiä. Heidän tuli tunnistaa tunnilla käytetyllä ohjelmointikielellä tehtyjä rakenteita (esimerkiksi setup- ja draw-funktiot), jotta kuviot ja mahdolliset toiminnot toimisivat tarkoituksen mukaisesti. Nämä edellä mainitut tavoitteet ovat mainittu e-perusteissa digitaalisen osaamisen kuvauksissa (Opetushallitus 2022).

Näin tarkasteluna yksi tapa miten luovan ohjelmoinnin opetuskokonaisuus voidaan toteuttaa lukion kuvataiteen opetuksessa, on rakentaa kokonaisuus, jossa opettajajohtoinen mallintaminen, yhteinen käsitteellistäminen ja itsenäinen kokeileva työskentely muodostavat yhtenevän kokonaisuuden. Lisäksi tutkimus osoittaa, että luovan ohjelmoinnin kokonaisuuden toteutus edellyttää huolellista ajallista, teknistä ja pedagogista suunnittelua. Tämän tyyppinen rakenne mahdollistaa tämän tutkimuksen valossa sen, että tekninen oppiminen sekä taiteellinen ilmaisu kehittyvät. Räsäsen (2008) näkökulma oli tämän tutkimuksen kannalta merkityksellinen, sillä myös kehittämäni luovan ohjelmoinnin opetuskokonaisuudessa oppiminen ymmärretään kokemuksellisenä, reflektiivisenä ja merkityksiä rakentavana prosessina, jossa tekninen työskentely kytkeytyy laajempiin kulttuurisiin ja yhteiskunnallisiin kysymyksiin.

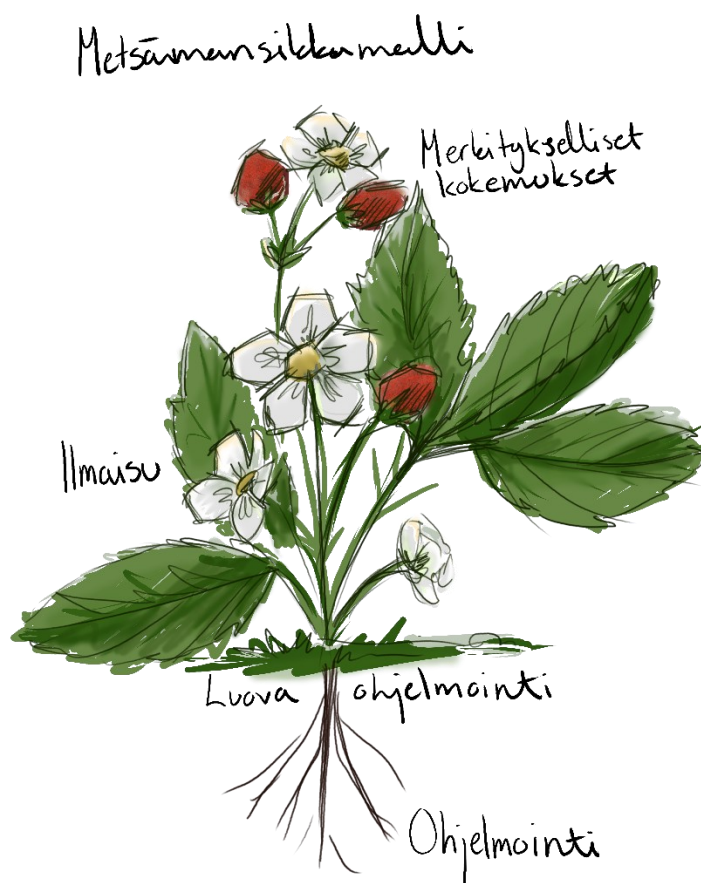
## 5.2 Miksi ohjelmointia?

Lukion opetussuunnitelma nimeää yhdeksi kuvataiteen oppiaineen tehtäväksi, että sen tulisi kannustaa uusien *välineiden, materiaalien ja ilmaisun keinojen kokeilemiseen sekä niiden luovaan soveltamiseen* ja mainitsee myös, että esimerkiksi teknologia on kuvailmaisun väline, mutta myös ilmiö, jota tutkia. (Opetushallitus 2021). Luova ohjelmointi toimii mielekkäänä tapana sekä tekemisen välineenä ja materiaalina sen lisäksi, että se avaa pohdintaa nykyhetken teknologian ilmiöistä. Lukio-opiskelijoiden tulkinnat koodista olivat osin linjassa Dufvan (2018) ja Dufvan & Dufvan (2016) koodin metaforien kanssa. Karnevaali tuli esiin erityisesti, kun koodi koettiin leikkikenttänä ja ilmaisun välineenä sekä materiaalina. Lisäksi opiskelijat pyrkivät teoksillaan luomaan keskustelua ja olivat pohtineet teoksen kautta ajankohtaisia aiheita. Kulttuurimetafora tuli esiin erityisesti silloin kun opiskelija pohti avoimen lähdekoodin sivustoja tai sosiaalista mediaa koodin kautta.

Tuloksia voidaan myös ymmärtää Deweyn (2010) taiteellisen ilmaisun käsitteen kautta, jossa taide ei rajaudu lopputulokseen vaan muodostuu tekemisen, materiaalin ja kokemuksen vuorovaikutuksessa. Opiskelijoiden työskentelyssä ohjelmointi ei ollut pelkkä väline valmiin teoksen tuottamiseen vaan prosessi, jossa kokeileminen, virheiden kautta eteneminen ja jatkuva muokkaaminen muodostivat keskeisen osan ilmaisua. Tämä tukee tutkimuksen havaintoa siitä, että

luova ohjelmointi vahvistaa käsitystä taiteellisesta ilmaisusta prosessina, jossa merkitykset rakentuvat toiminnan aikana eikä vasta valmiissa teoksessa. Deweyn (2010) ajatus rajoitteiden merkityksestä taiteelliselle ilmaisulle auttaa ymmärtämään myös koodin roolia pedagogisessa kokonaisuudessa. Ohjelmointi ei toiminut opiskelijoille vapaana ilmaisun leikkikenttänä vaan osin rajoittavana materiaalina, joka kuitenkin mahdollisti uudenlaisia ratkaisuja ja oivalluksia. Näin koodi näyttyy Deweyn (2010) tarkoittamassa mielessä ilmaisun ehtona, ei sen esteenä. Tämä selittää osaltaan sen, miksi lukio-opiskelijat kuvasivat työskentelyä sekä haastavana että palkitsevana.

Aineistossa on havaittavissa piirteitä transformatiivisesta oppimisesta, sillä opiskelijoiden käsitykset taiteesta, luovuudesta, teknologiasta ja omasta osaamisesta näyttävät osin laajentuneen opetuskokonaisuuden aikana (Anttila 2022, 1:3). Kyse ei kuitenkaan välttämättä ole kaikilta osin täydellisestä transformatiivisesta oppimisesta vaan pikemminkin näkökulmien asteittaisesta uudelleen jäsentymisestä.



Kuva 17. Metsämansikkamalli, Jenna Hepoaho 2026

Metsämansikkamalli visualisoi luovan ohjelmoinnin mahdollisuuksia (Kuva 17). Tuloksiini perustuen näen luovan ohjelmoinnin kasvualustana, joka mahdollistaa luovuuden kukkimisen ja merkitysten sekä uuden oppimisen, kypsymisen ja uudelleenkasvun. Luova ohjelmointi toimii alustana, jossa taide, teknologia ja oppiminen kohtaavat. Se on turvallinen tila tutkia ja ilmaista. Ohjelmointi toimii tämän tutkimuksen mukaan luovan ohjelmoinnin perustana, jossa koodin lukutaito, ohjelmoinnillinen ajattelu ja tekninen ymmärrys muodostavat luovan ohjelmoinnin juuriston. Varsi kuvaa prosessia ja kasvua, joka muodostuu kokeilusta, leikistä, kärsivällisyydestä ja sinnikkyydestä. Se on luovuuden kasvua uusien mahdollisuuksien seurauksena. Kukkat kuvaavat taiteellisen ilmaisun näkyvyyttä aistillisin teoksin. Mansikan siemen kuvaa merkityksellisiä kokemuksia, jotka seurauksena tapahtui transformatiivista oppimista, joka mahdollistaa ajatusten muutoksen ja uudelleenkasvun. Metsämansikkamalli kuvaa myös rajallisuutta, mansikka ei voi olla suuri mänty vaan sen täytyy toimia sille annettujen rajojen sisällä.

Tutkimuskysymykseen vastaten voidaan todeta, että luovan ohjelmoinnin opetuskokonaisuus voidaan toteuttaa kuvataiteen opetuksessa pedagogisesti toimivana kokonaisuutena, kun siinä yhdistyvät opettajajohtoinen ja kokeileva työskentely ja kriittinen pohdinta. Tämän tutkielman perusteella se tuottaa opiskelijoille uudenlaisen taiteellisen ilmaisun välineen, vahvistaa käsitystä taiteesta prosessina sekä avaa kriittisen näkökulman digitaaliseen kulttuuriin. Samalla tulokset osoittavat, että luova ohjelmointi ei ainoastaan laajenna taiteen opetuksen menetelmiä vaan myös tuo näkyväksi taiteellisen ilmaisun rajoja (rajattomuutta) post-digitaalisessa kontekstissa, jossa teknologia, kokemus ja merkityksenmuodostus liimautuvat erottamattomasti toisiinsa.

## 6 Pohdinta

Tämän tutkimuksen luotettavuutta voidaan tarkastella ensisijaisesti sen aineistollisen ja pedagogisten rajoitteiden kautta. Tutkimus toteutettiin rajatussa opetuskokonaisuudessa, jossa opetuksen aikataulu osoittautui tiukaksi jo suunnitteluvaiheessa ja sitä jouduttiin edelleen tarkentamaan toteutuksen aikana. Tämä vaikutti siihen, että osa sisällöistä jäi väistämättä suppeammaksi kuin olisi ollut pedagogisesti ideaalista. Erityisesti generatiivisen taiteen laajempien esimerkkien käsittelyn osalta olisi ollut tärkeää, että aikaa olisi ollut enemmän. Lisäksi tekniset haasteet ja oppituntien aikarajoitteet vähensivät aikaa yhteiselle pohdinnalle ja syvemmälle keskustelulle, mikä saattoi rajata aineison monipuolisuutta ja syvällisyyttä.

Tutkimuksen luotettavuuteen vaikuttaa myös asemani opintojakson opettajana. Olin ehtinyt ennen luovan ohjelmoinnin kokonaisuutta tulemaan tutuksi lukio-opiskelijoille ja he minulle. Huomioitavaa on myös, että oman osaamisen lähtökohta ohjelmoinnille oli vielä kehittyvä ja kokeileva, mikä toisaalta mahdollisti toisaalta opiskelijoiden näkökulman ymmärtämisen, mutta saattoi samalla rajoittaa pedagogista varmuutta erityisesti tilanteissa, joissa opiskelijan osaamistaso ylitti oman teknisen osaamisen.

Koodin syvempi ymmärtäminen esimerkiksi Dufvan (2018) tarkoittamassa merkityksessä jäi mielestäni hieman latteaksi tekemisen aikana. Enkä voi tietää ymmärsivätkö lukio-opiskelijat täysin koodin luonnetta. Toisaalta tätä latteutta paikkasi mielenkiintoiset Padlet -keskustelut, joiden aikana nousseita aiheita selvästi käsiteltiin myöhemmin teoksissa ja reflektioteksteissä. Koodilla leikkiminen jäi osin suppeaksi. Useampi noudatti tehtävänantoa ja pyrki muokkaamaan yhdessä kirjoitettua esimerkkikoodia. Osa taas ei muokannut koodia ollenkaan. Tulevaisuudessa olisikin mielekästä testata eri ohjelmointiympäristöjä.

Yleistettävyyden näkökulmasta tutkimuksen tuloksia ei voida pitää sellaisinaan laajasti siirrettävänä kaikkiin lukion kuvataiteen opetuksen konteksteihin. Aineisto muodostui rajatusta lukio-opiskelijaryhmästä, joka koostui valinnaisen kuvataiteen opiskelijoista. Näin voidaan olettaa, että valtaosalla oli jo valmiiksi kiinnostusta taideaineisiin. Lisäksi luovan ohjelmoinnin jakso sijoittui osaksi nykytaiteen opetuskokonaisuutta, mikä tarkoittaa, että opiskelijoiden käsitykset taiteesta ja teknologiasta saattoivat kehittyä jo ennen ohjelmointiosuutta. Näin ollen ei voida yksiselitteisesti todeta, missä määrin havaitut muutokset johtuvat juuri ohjelmoinnin tuntikokonaisuudesta ja missä määrin yleisesti opintojaksosta. Tästä huolimatta tulokset antavat viitteitä siitä, että luovalla ohjelmoinnilla voi olla potentiaalia taiteellisen ilmaisun kehittymisessä ja laajentamisessa sekä opiskelijoiden digitaalisen kulttuurin kriittisen ymmärryksen tukemisessa. Erityisesti opiskelijoiden omat kokemukset oman pystyvyyden vahvistumisesta sekä ennakoasenteiden muuttumisesta viittaavat siihen, että menetelmällä voi olla pedagogista merkitystä myös laajemmin tarkasteltuna, vaikka sen vaikutuksia ei voida tässä tutkimuksessa yleistää varmuudella.

Tarkoitukseni ei ole latistaa tuloksia vaan tarkastella erityisen kriittisesti omaa toimintaani. Tutkimus onnistui tuomaan mielenkiintoisia havaintoja siitä, miten lukio-opiskelijat kokivat luovan ohjelmoinnin kokonaisuuden. Se oli oppimisen kannalta rikasta ja moninaista. Tulokset kertoivat, että lukio-opiskelijat eivät ainoastaan oppineet ohjelmointia vaan oppivat myös itsestään ja laajensivat käsitystään taiteesta. Luova ohjelmointi toimii luovana leikkikenttänä, joka vaatii sinnikkyyttä, mutta samalla antaa onnistumisen kokemuksia. Tutkimuksen tuoman kokemuksen

myötä ja ottaen huomioon Opetushallituksen digitaalisen osaamisen kuvaukset voim kannustaa tuomaan luovaa ohjelmointia myös osaksi perusopetuksen kuvataiteen opetusta.

Jatkokehittämisen näkökulmasta tutkimus nostaa esiin useita keskeisiä suuntia. Ensinnäkin opetuskokonaisuuden rakenteellista jäsentelyä olisi perusteltua kehittää siten, että yksittäisille teemoille jäisi enemmän aikaa ja syventymismahdollisuuksia. Nykyisessä toteutuksessa sisällöllinen laajuus saattoi osin tapahtua syvällisen käsittelyn kustannuksella. Tulevissa toteutuksissa voisi olla perusteltua keskittyä vähempään määrään keskeisiä teemoja ja rakentaa niiden ympärille enemmän kokeilevaa portfoliomaista työskentelyä. Toiseksi jatkotutkimukseksi olisi mielekästä tarkastella luovaa ohjelmointia omana pidempänä opintojaksonaan, jossa opiskelijoilla olisi enemmän aikaa kehittää omaa ilmaisuaan. Teknologisen ulottuvuuden laajentaminen esimerkiksi Arduino- tai Raspberry Pi-pohjaisiin kokeiluihin voisi avata uusia näkökulia taiteen, materiaalisuuden ja koodin suhteen. Kolmanneksi voidaan nostaa vielä kysymys siitä, miten luovaa ohjelmointia voisi lähestyä enemmän kokeilevuuden kautta. Esimerkiksi Cables -ympäristö voisi antaa enemmän mahdollisuuksia keskittyä kokeilevuuteen. Siinä ei tarvitse osata kirjoittaa koodia, vaan se on vähän saman tyyppinen kuin Scratch. Tämän tutkimuksen valossa kyseenalaistan hieman sitä, onko tekstipohjainen ohjelmointi paras tapa lähestyä luovaa ohjelmointia. Toki voisin hypotesina asettaa sen, että tekstipohjainen ohjelmointi tuo paremmin näkyväksi sen, miten eri laitteet ja ohjelmat toimivat.

Tällä hetkellä pinnalla oleva Opetushallituksen Uudet lukutaidot -kehittämiprojektin myötä luovaa ohjelmointia voitaisiin lähteä pohtimaan myös tämän kautta. Siihen voisi yhdistää Dufvan & Dufvan (2016) mainitseman koodin lukutaidon. Näen, että ohjelmoinnillisella ajattelulla ja koodin lukutaidolla on paljon yhteneväisyyksiä. Millä tavoin luova ohjelmointi tukee ohjelmoinnillisen ajattelun kehitystä ja koodin lukutaitoa?

Kiitos Lapin yliopiston kuvataidekasvatuksen koulutusohjelmalle mahdollisuudesta tutustua luovaan ohjelmointiin. Lapin yliopiston kuvataidekasvatuksen koulutusohjelmaan on sisällytetty maisteritutkintoon kurssi, jossa tavoitteena on pelillisyyden ja ohjelmoinnin soveltaminen osana kuvataidekasvatusta (Lapin yliopisto 2024). Olisi mielenkiintoista saada lisää tutkimusta, miten kuvataidekasvatuksen opiskelijat kokevat luovan ohjelmoinnin, sillä Knochel & Patton (2015) puhuvat taideopiskelijoiden ohjelmoinnillisen ajattelun tärkeydestä. Kuvataidekasvatuksen opiskelijat ovat kuitenkin niitä, jotka siirtyvät opettamaan kuvataidetta ja mahdollisesti tuomaan uutta osaamista kuvataiteen opetuksen kentällä.

## 7 Lähteet

Anttila, E. (toim). (2022). *Ihmis- ja oppimiskäsitykset taideopetuksessa*. Taideyliopiston Teatterikorkeakoulu, Helsinki: Teatterikorkeakoulun julkaisusarja 58, [URN:NBN:fi-fe20201221101868](https://nbn-resolving.org/urn:nbn:fi-fe20201221101868)

Brown, N. C. C., & Wilson, G. (2018). *Ten quick tips for teaching programming*. *PLOS Computational Biology*, 14(4), e1006023-. <https://doi.org/10.1371/journal.pcbi.1006023>

Chibalashvili, A., Savchuk, I., Olianina, S., Shalinskyi, I., & Korenyuk, Y. (2023). *Creative Coding as a Modern Art Tool*. *BRAIN. Broad Research in Artificial Intelligence and Neuroscience*, 14(2), 115–127. <https://doi.org/10.18662/brain/14.2/447>

Cramer, F. (2014). *What Is 'Post-Digital'? A Peer-Reviewed Journal About*, 3(1), 10–24. <https://doi.org/10.7146/aprja.v3i1.116068>

Dewey J. (1934/2010). *Taide kokemuksena*. Julkaistu englanniksi 1934. Suomentaneet Antti Immonen ja Jarkko S. Tuusvuori 2010. Niin&näin, Tampere.

Dufva, T. (2018). *Art education in the post-digital era - Experiential construction of knowledge through creative coding*. <https://aaltodoc.aalto.fi/items/0b7bbd6f-fa55-4ba9-aebb-919538dd0940>

Dufva, T., & Dufva, M. (2016). *Metaphors of code—Structuring and broadening the discussion on teaching children to code*. *Thinking Skills and Creativity*, 22. <https://doi.org/10.1016/j.tsc.2016.09.004>

Dufva, T., & Dufva, M. (2019). *Grasping the future of the digital society*. *Futures*, 107, 17–28. <https://doi.org/10.1016/J.FUTURES.2018.11.001>

Fagerlund, J. (2022). *Tietokonevallankumous ja ohjelmoinnillinen ajattelu peruskoulussa – Havaintoja mikro- ja makrotasoilta*. *Kasvatus & Aika*, 16(1). <https://doi.org/10.33350/ka.111888>

Fuentes Martinez, A. (2024). *Practice beyond technology when programming and mathematics teaching converge*. University West, Division of Media and Design. <https://urn.kb.se/resolve?urn=urn:nbn:se:hv:diva-21050>

Greenberg, I., Kumar, D., & Xu, D. (2012). Creative coding and visual portfolios for CS1. Proceedings of the 43rd ACM Technical Symposium on Computer Science Education, SIGCSE '12, 247–252. <https://doi.org/10.1145/2157136.2157214>

Hakala, J. T. (2024). Laadullisen tutkimuksen ABC : menetelmäopas opinnäytteen tekijälle. Gaudemus.

Hakala, J. T. (2024). Laadullisen tutkimuksen ABC : menetelmäopas opinnäytteen tekijälle. Gaudemus. E-kirja.

Heikkinen, H. L. T., Kaukko, M., Friman, M., Salo, P., Kiilakoski, T., Huttunen, R., . . . Kemmis, S. (2023). *Toimintatutkimus: Käytännön opas*. Vastapaino. E-kirja.

Jokela, T., & Huhmarniemi, M. (2025). *Taideperustainen toimintatutkimus kuvataidekasvatuksen käytäntöjen kehittämisen välineenä*. *Research in Arts and Education*, 2025(2), 100–125. <https://doi.org/10.54916/rae.145624>

Jokela, T. & Huhmarniemi, M. (2020) *Taideperustainen toimintatutkimus soveltavan taiteen kehittämisen välineenä*. <https://urn.fi/URN:NBN:fi-fe2020050424787>

Juhila, K. (2021). *Laadullisen tutkimuksen ominaispiirteet*. Teoksessa Jaana Vuori (toim.) *Laadullisen tutkimuksen verkkokäsikirja*. Tampere: Yhteiskuntatieteellinen tietoaarkisto [ylläpitäjä ja tuottaja]. <https://www.fsd.tuni.fi/fi/palvelut/menetelmaopetus>

Juntunen, M-L. (2011) *Liike, rytmi ja musiikki: Jaques-Dalcrozen pedagogista perintöä jäljittämässä*. Teoksessa: *Taiteen jälki: Taidepedagogiikan polkuja ja risteyksiä*. <https://urn.fi/URN:ISBN:978-952-9765-64-5>

Kiasma. (n.d.). Nykytaiteen sanasto. Verkkosivu. <https://kiasma.fi/nykytaiteen-sanasto/>

Knochel, A. D., & Patton, R. M. (2015). *If Art Education Then Critical Digital Making: Computational Thinking and Creative Code*. *Studies in Art Education*, 57(1). <https://doi.org/10.1080/00393541.2015.11666280>

Lapin yliopisto. (2024). *UKUV0118V24 opintojakso* <https://opinto-opas-lay.peppi4.lapit.csc.fi/fi/opintojakso/UKUV0118V24/28882>

MAOL ry. (n.d.). *Taidetta ohjelmoimalla*. Verkkosivu. <https://maol.fi/materiaalit/taidetta-ohjelmoimalla/>

Mediataitokoulu. (2026). *Uudet lukutaidot*. Verkkosivu. <https://mediataitokoulu.fi/avainsanat/uudet-lukutaidot/>

Mehackit Oy. (n.d.). *Kuvataide ja ohjelmointi Processingilla*. Verkkosivu. [https://mehackit.org/kurssit/kuvataide\\_ja\\_ohjelmointi\\_processingilla/](https://mehackit.org/kurssit/kuvataide_ja_ohjelmointi_processingilla/)

Mertala, P., Palsa, L., & Slotte Dufva, T. (2020). *Monilukutaito koodin purkajana: Ehdotus laajalaiseksi ohjelmoinnin pedagogiikaksi*. *Media & viestintä*, 43(1). <https://doi.org/10.23983/mv.91079>

Mäkitalo K., Tedre M., Laru J., Valtonen T., Iwata M. & Koivisto J. (2024). Learning computational thinking in co-creation projects with modern educational technology: experiences from Finland. Teoksessa: *Computational thinking curricula in K-12: international implementations*. The MIT Press.

Nojonen, T. (2024). *Ohjelmoinnin opetus lukioden paikallisissa opetussuunnitelmissa*. [https://www.utupub.fi/bitstream/handle/10024/176665/Nojonen\\_Gradu\\_Valmis.pdf;jsessionid=F42718F9C9428C61E6AA2ED416B9590F?sequence=1](https://www.utupub.fi/bitstream/handle/10024/176665/Nojonen_Gradu_Valmis.pdf;jsessionid=F42718F9C9428C61E6AA2ED416B9590F?sequence=1)

Nousiainen, S., & Kivistö, A. (2022). Ohjelmoinnin opetuksen arviointi lukiokoulutuksessa.

Opetushallitus. (2014). *Perusopetuksen opetussuunnitelman perusteet*. Verkkosivu. [https://www.oph.fi/sites/default/files/documents/perusopetuksen\\_opetussuunnitelman\\_perusteet\\_2014.pdf](https://www.oph.fi/sites/default/files/documents/perusopetuksen_opetussuunnitelman_perusteet_2014.pdf)

Opetushallitus. (2019). *Lukion opetussuunnitelman perusteet*. Verkkosivu. <https://eperusteet.opintopolku.fi/eperusteet-service/api/dokumentit/10026171>

Opetushallitus. (2023) *Digitaalisen osaamisen kuvaukset – Tukea perusteiden toimeenpanoon*. Verkkosivu. <https://eperusteet.opintopolku.fi/#/fi/digiosaaminen/8706410/osaamiskokonaisuuspaaalue/8715977>

P5.js. (2026). Verkkosivu. <https://p5js.org/>

Koski, L. (2020) *Teksteistä teemoiksi. Dialoginen tematisointi*. Teoksessa: Puusa, A., Juuti, P., & Aaltio, I. *Laadullisen tutkimuksen näkökulmat ja menetelmät*. Gaudeamus. E-kirja.

Ranta, V. (2020). *Kuvis: Heart: Ohjelmointi: Opettajien kokemuksia kuvataiteen ja ohjelmoinnin yhdistämisestä peruskoulussa*. Aalto yliopisto. <https://urn.fi/URN:NBN:fi:aalto-2020121256171>

Reas, Casey., & Fry, Ben. (2007). *Processing : a programming handbook for visual designers and artists*. MIT Press.

Räsänen, M. (2008). *Kuvakulttuurit ja integroiva taideopetus*. Taideteollinen korkeakoulu.

Räsänen M., (2011). *Taiteet kognition ja kulttuurin kentällä*. Teoksessa: Taiteen jälki: Taidepedagogiikan polkuja ja risteyskiä. <https://urn.fi/URN:ISBN:978-952-9765-64-5>

Saastamoinen, R. (2011) *Prologi: Virhe taidepedagogiikan tehtävänä*. Teoksessa: Taiteen jälki: Taidepedagogiikan polkuja ja risteyskiä. <https://urn.fi/URN:ISBN:978-952-9765-64-5>

Saethre-McGuirk, E. M. (2022). *Making Things and Teaching the Creative Arts in the Post-Digital Era; Seeing and Experiencing the Self and the Object through a Digital Interface*.

Slotte Dufva, T. (2021). *Creative Coding as Compost(ing)*. Teoksessa: *Post-Digital, Post-Internet Art and Education*. [https://doi.org/10.1007/978-3-030-73770-2\\_16](https://doi.org/10.1007/978-3-030-73770-2_16)

Terroso, T., & Pinto, M. (2022). *Programming for Non-Programmers: An Approach Using Creative Coding in Higher Education*. Teoksessa: A. Simões & J. C. Silva (Toim.), *Third International Computer Programming Education Conference (ICPEC 2022)* (Vsk. 102, ss. 13:1–13:8). Schloss Dagstuhl – Leibniz-Zentrum für Informatik. <https://doi.org/10.4230/OASlcs.ICPEC.2022.13>

Tieteen termipankki. (2026). *Nykytaide*. Verkkosivu. <https://tieteentermipankki.fi/wiki/Taidehistoria:nykytaide>

Wing, J. M. (2006). *Computational thinking*. *Commun. ACM*, 49(3), 33–35. <https://doi.org/10.1145/1118178.1118215>

Wood, Z. J., Muhl, P., & Hicks, K. (2016). *Computational Art: Introducing High School Students to Computing via Art*. *Proceedings of the 47th ACM Technical Symposium on Computing Science Education, SIGCSE '16*, 261–266. <https://doi.org/10.1145/2839509.2844614>

## 8 Liitteet

### 8.1 Liite 1. Padlet -kysymykset

1. Tunti: Onko koodi väline vai osa taidetta? Kuka on tekijä, jos koodi on avointa ja muokattavissa?
2. Tunti: Miten algoritmit tuottavat sattumaa: Algoritmit ohjaavat somefeedejä ja mainoksia – onko se oikeasti sattumaa vai jonkun suunnittelemaa?
3. Tunti: Hiiri, näppäimistö, kosketus... Kuka ohjaa ketä? Onko käyttäjä aina se, joka päättää? Millä tavoin interaktiiviset järjestelmät (pelit, some, sovellukset) ohjaavat käyttäjää toimimaan tietyllä tavalla?

### 8.2 Liite 2. Opiskelijoiden reflektointitehtävän apukysymykset

Miten valitsemasi teema liittyy “koodattuun maailmaan”?

Mitä halusit teoksellasi kertoa tai tuoda esiin?

Mikä kohta koodissa oli erityisen kiinnostava tai haastava?

Mitä uutta oivalsit koodatessasi?

Oliko teoksessasi osia, jotka syntyivät vahingossa, mutta päätit säilyttää?

Jos tekisit teoksen uudelleen, mitä muuttaisit?

Miten teoksesi voisi liittyä laajemmin teknologiaan tai digitaaliseen kulttuuriin?

Mistä olet erityisen ylpeä?

Mikä oli sinulle uutta tai yllättävää?

Mitä opit itsestäsi luovana tekijänä tässä prosessissa?

Millaista oli tehdä taidetta koodin avulla verrattuna perinteisiin menetelmiin?

### 8.3 Liite 3. Tutkimuslupa

Tutkimuslupa

Hei sinä Nykytaide -opintojakson opiskelija,

Olen Jenna Hepoaho, Lapin yliopiston kuvataidekasvatuksen maisteriopiskelija. Kerään aineistoa pro gradu -tutkimukseeni Lohjan yhteislyseon lukion Nykytaide -opintojaksolta. Tutkimukseni liittyy luovaan koodaukseen osana lukion kuvataiteen opetusta. Pyydän sinua osallistumaan tutkimukseeni. Osana tutkimustani kerään aineistoa Nykytaide -opintojaksolla, ja aineisto muodostuu oppitunneilla kerätyistä materiaaleista eli omista havainnoistani ja muistiinpanoistani, opiskelijoiden tekemistä teoksista ja kyselyvastauksista. Aineistoa tai sen osia voidaan raportoida, esittää tai julkaista valmiissa pro gradu -tutkimuksessani. Aineistossa esiintyvät henkilöt pidetään anonyyminä eli nimettömänä, ja aineisto säilytetään vain tätä tutkimusta varten.

Aineistona käytettävät kyselyt ja työt ovat osa Nykytaide-opintojakson arvioitavia töitä, eikä näin ollen tuota lisätöitä opintojakson aikana. Osallistumisen peruminen tai osallistumatta jättäminen ei vapauta tutkimukseen liittyvän tuntikokonaisuuden töistä.

Ystävällisin terveisin,

Jenna Hepoaho

[jhepoaho@ulapland.fi](mailto:jhepoaho@ulapland.fi)