



OULUN YLIOPISTO  
UNIVERSITY of OULU

Koivisto Jussi Oskari

Minecraft ohjelmointiympäristönä ohjelmoinnin alkeiden opetuksessa

Kasvatustieteen kandidaatintyö

KASVATUSTIETEIDEN TIEDEKUNTA

Teknologiapainotteinen luokanopettajakoulutus



Luokanopettajankoulutus		Tekijä/Author	
		Koivisto Jussi	
Työn nimi/Title of thesis			
Minecraft ohjelmointiympäristönä ohjelmoinnin alkeiden opetuksessa			
Pääaine/Major subject	Työn laji/Type of thesis	Aika/Year	Sivumäärä/No. of pages
Kasvatustiede	Kandidaatintyö	2016	22
Tiivistelmä/Abstract			
<p>Ohjelmointi on otettu osaksi pakollista perusopetusta ja sitä aletaan opettaa uusien perusopetuksen opetussuunnitelman perusteiden myötä vuonna 2016. Ohjelmointia on opetettu peruskouluissa aikaisemminkin, mutta opetusta vähennettiin runsaasti. Ohjelmointi oli vaikeaa eikä se kiinnostanut. Ohjelmointi ei ole juurikaan muuttunut, mutta ohjelmoinnin opetukseen tulee panostaa, ettei toisteta samoja virheitä kuin 80- ja 90-luvun ohjelmoinnin opetuksessa.</p> <p>Tietotekniikan kehittyessä erilaiset laitteet ja niillä pelattavat pelit ovat kasvattaneet suosiotaan. Yksi suosituimmista peleistä on tietokoneella, tableteilla ja eri pelikonsoleilla pelattava Minecraft. Minecraft on pelinä hyvin yksinkertainen, mutta sen avoin maailma ja modaus-mahdollisuudet tekevät siitä hyvin monimuotoisen pelin, jossa on potentiaalia myös opetuskäyttöä ajatellen.</p> <p>Tutkielmani on kirjallisuuskatsaus, jossa tarkastelen ohjelmoinnin opetusta ohjelmoinnin oppimisen haasteiden kautta. Tutkin, mitä haasteita ja esteitä ohjelmoinnin oppimisessa kohdataan ja kuinka Minecraftin tarjoamassa virtuaalisessa oppimisympäristössä voidaan vastata kyseisiin haasteisiin.</p> <p>Ohjelmoinnin oppimisen haasteiksi nousivat ohjelmoinnin irtonaisuus arjesta ja fyysisen laitteen ongelma, ratkaisun esittäminen tietokoneen ymmärtämässä muodossa, häiriötekijöiden minimointi ja helppokäyttöisyys sekä motivointi. Aikaisemmat tutkimukset osoittavat, että Minecraftilla voidaan oppimisympäristönä vastata pääosin ohjelmoinnin oppimisen haasteisiin, erityisesti motiivoinnin osalta. Tällä hetkellä tutkimusta on tehty varsin vähän, mutta aiheessa on vahvasti potentiaalia jatkotutkimukselle. Tarkoitukseni on jatkaa aiheen tutkimista pro gradu -tutkielmassa.</p>			
Asiasanat/Keywords Ohjelmointi, Ohjelmointiympäristö, Minecraft			

## Sisältö

<b>Johdanto</b> .....	<b>1</b>
<b>1 Tutkimuksen tavoite ja tutkimuskysymykset</b> .....	<b>3</b>
<b>2 Ohjelmointiin ja sen oppimiseen liittyviä käsitteitä</b> .....	<b>4</b>
2.1 Ohjelmoinnin opetus perusopetuksen opetussuunnitelman perusteissa 2014 .....	4
2.2 Graafinen ohjelmointiympäristö .....	5
2.3 Algoritminen ajattelu .....	7
<b>3 Ohjelmoinnin oppimisen haasteita</b> .....	<b>8</b>
3.1 Haaste 1: Irtonaisuus arjesta ja fyysisen laitteen ongelma .....	8
3.2 Haaste 2: Ratkaisun esittäminen tietokoneen ymmärtämässä muodossa .....	8
3.3 Haaste 3: Häiriötekijöiden minimointi ja helppokäyttöisyys .....	9
3.4 Haaste 4: Motivointi .....	10
<b>4 Minecraft oppimisympäristönä ohjelmoinnin opetuksessa</b> .....	<b>11</b>
4.1 Mikä on Minecraft?.....	11
4.2 Minecraft opetuskäytössä ja oppimisympäristönä .....	12
4.3 Minecraft ohjelmoinnin opetuksessa ja algoritmisen ajattelun edistäjänä .....	12
4.3.1 Redstone eli punakivi.....	12
4.3.2 Ohjelmoitavat kilpikonnat - ComputerCraft ja ComputercraftEdu.....	13
4.3.3 Muita tapoja opettaa ohjelmointia Minecraftissa .....	14
<b>5 Aiheesta tehty aiempi tutkimus</b> .....	<b>15</b>
5.1 Tutkimus A: ComputerCraft - innostava ohjelmointiympäristö .....	15
5.2 Tutkimus B: Ohjelmoinnin oppimisesta hauskaa ja helppoa CodeBlocksin avulla .....	16
5.3 Tutkimus C: Aiheesta kiinnostuminen Minecraftin avulla .....	17
5.4 Tutkimus D: Teksti- vai kuvakepohjainen ohjelmointi?.....	17
<b>6 Johtopäätökset ja pohdinta: Kuinka Minecraft vastaa ohjelmoinnin oppimisen haasteisiin?</b> .....	<b>19</b>
<b>Lähdeluettelo:</b> .....	<b>23</b>



## Johdanto

Perusopetuksen opetussuunnitelman 2014 julkaistussa versiossa ohjelmointi otetaan osaksi perusopetusta (POPS 2014). Tietokoneisiin perehtyminen ja ohjelmoinnin opetus eivät kuitenkaan ole aina olleet osa kaikille yhteistä perusopetusta. Vuonna 2004 julkaistussa Perusopetuksen opetussuunnitelmassa tietokoneiden käyttöön liittyvä opetussisältö oli hyvin rajallista. Kyseisessä versiossa opetussuunnitelmasta sanat ”tietokone” ja ”tietotekniikka” esiintyvät vain viestinnän ja tekstin tuottamisen yhteydessä eli osana äidinkielen opetusta, sekä käsiteltäessä yläkoulun valinnaisainetta, tietotekniikkaa (POPS 2004). Vuonna 2014 julkaistussa versiossa ohjelmointi on jo selkeästi osana perusopetuksen oppimistavoitteita (POPS 2014). Suomi ei kuitenkaan ole ainoa maa, jossa on kiinnostuttu ohjelmoinnin opettamisesta lapsille. Tästä osoituksena maailmanlaajuinen ”Hour of code” projekti, joka rohkaisee ihmisiä ympäri maailman tutustumaan ohjelmointiin (Hour of Code 2015).

Ohjelmointi ei ole uusi asia, sillä niin kauan kuin on ollut tietokoneita, on ollut myös ohjelmointia. Miksi ohjelmoinnin osaaminen on juuri nyt ajankohtaista? Tähän kysymykseen täytyy etsiä vastausta 1980-luvulta, jolloin tietotekniikka tuotiin ensimmäistä kertaa osaksi perusopetusta. Tietotekniikkaa oli opetettu lukioissa 1980-luvun alussa hyvin ohjelmointi painotteisesti. Ohjelmoinnin huomattiin kuitenkin turhauttavan monia, ja kun tietotekniikka tuotiin osaksi perusopetusta, ei haluttu tehdä samoja virheitä kuin lukion tietotekniikan opetuksessa oli tehty. Kun vuonna 1985 tietotekniikan opetusta alettiin tuoda perusopetukseen, ohjelmoinnin osuutta oli vähennetty ja tulevana vuosina vähennettiin vielä enemmän. Näytti siltä, että mitä vähemmän ohjelmointia opetettiin osana tietotekniikan opetusta, sitä enemmän tietotekniikka alkoi kiinnostaa. Tuolloin ohjelmoinnissa käytettiin paljon kritisoitua BASIC-kieltä ja ohjelmoinnin opetus jäi hyvin tekniselle tasolle. Vaikeaselkoisuus ja matemaattisuus nousivat oppimisen esteiksi. 1990-luvulle siirryttäessä ATK-opetusta järjestettiin myös alakouluissa. Tuolloin ATK-opetus oli lähinnä hyödyllisten ohjelmien ja opetusohjelmien käyttöä. Ohjelmointia opetettiin enää valinnaisena aineena yläkouluilla. Myöhemmin 2000-luvulla tietokonekerhojen toiminta väheni huomattavasti, kun tietokonepelit tekivät läpimurron. Lisäksi kerhojen hyöty väheni apuohjelmien ja helppokäyttöisyyden yleistyessä. (Saarikoski 2006)

Ohjelmoinnin opetus ei siis ole tulossa uutena peruskouluun, vaan se on tekemässä paluuta osaksi perusopetusta. Nyt kun ohjelmointi on taas noussut pinnalle ja perusopetukseen, on aika katsoa menneisyyteen ja oppia siitä. Miten ohjelmoinnin opetuksessa onnistuttaisiin tällä kertaa? Miten ohjelmoinnin oppimisesta voisi tehdä tarpeellista, mielenkiintoista, helppoa ja nautittavaa? Jos tietokonepelit veivät opetuksellisten tietokonekerhojen suosion, voisivatko ne olla apu suosion palauttamisessa?

Opetan itse ohjelmointia Suomen Koodikoulu nimisessä yrityksessä. Olen siis itse todistamassa uutta “koodivillitystä”, jossa ohjelmoinnin opettamista lapsille nostetaan korokkeelle. Vanhemmat ovat lisääntyvässä määrin kiinnostuneet laittamaan lapsensa ohjelmointikerhoihin opettelemaan ohjelmointia. Olen jo tehnyt kokeiluja Minecraftin käytöstä ohjelmoinnin opetuksessa, mutta varsinaista tutkimusta aiheesta en ole vielä tehnyt.

# 1 Tutkimuksen tavoite ja tutkimuskysymykset

Tutkimuksen tavoitteena on tutkia Minecraft-videopelin käyttöä ohjelmoinnin oppimisympäristönä peruskoulun ohjelmoinnin opetuksessa. Tutkielmani sisällöllä pyrin vastaamaan mahdollisimman tarkasti ja kattavasti seuraaviin kysymyksiin, jotka toimivat tutkielman tutkimuskysymyksinä:

I) Mikä tekee ohjelmoinnin oppimisesta haastavaa?

II) Miten Minecraft oppimisympäristönä tukee ohjelmoinnin oppimista?

Ensimmäinen tutkimuskysymys ”Mikä tekee ohjelmoinnin oppimisesta haastavaa?” toimii teoreettisen viitekehysten pohjana. Tarkastelen, millaisia vaatimuksia voidaan asettaa oppimisympäristölle ja ohjelmoinnin oppimiselle ja kuinka ne voivat tehdä ohjelmoinnin oppimisesta haastavaa. Tarkoitukseni on tutkia peruskouluikäisten oppimista, mutta rajaan aiheen tässä vaiheessa vain aloitteleviin ohjelmoinnin oppijoihin.

Toinen kysymys ”Miten Minecraft oppimisympäristönä tukee ohjelmoinnin oppimista?” on tutkielmani tärkein kysymys. Saman kysymyksen alla analysoin Minecraftin tarkoituksenmukaisuutta oppimisympäristönä. Tutkimuskysymysten vuoropuhelulle jää tilaa, sillä jotta Minecraft oppimisympäristönä tukisi ohjelmoinnin oppimista, tulee sen käytöllä pystyä vastaamaan ohjelmoinnin oppimisen haasteisiin.

Tutkielman teoreettinen viitekehys on jaettu kahteen osioon, joissa on erotettu ohjelmoinnin opetus ja Minecraft ohjelmointiympäristönä käsiteltäväksi erikseen. Ohjelmoinnin opetusta käsittelevässä kappaleessa tarkkailen ohjelmoinnin opetustavoitteita Perusopetuksen opetussuunnitelman perusteissa 2014, ja tutkin ohjelmoinnin oppimisen haasteita. Toisessa kappaleessa tarkastelen Minecraftia oppimisympäristönä tutkien millaisia oppimista edistäviä elementtejä kyseinen videopeli sisältää, erityisesti ohjelmoinnin oppimiseen liittyen. Pyrin löytämään ohjelmoinnin opetukseen liittyvät haasteet ja kuinka niihin voidaan vastata Minecraftin käytöllä oppimisympäristönä.

## 2 Ohjelmointiin ja sen oppimiseen liittyviä käsitteitä

### 2.1 Ohjelmoinnin opetus perusopetuksen opetussuunnitelman perusteissa 2014

Perusopetuksen opetussuunnitelman perusteissa 2014 määräykset ja ohjeet on alakoulun osalta jaettu vuosiluokittain kahteen osioon: Vuosiluokat 1-2 ja vuosiluokat 3-6. Lisäksi on asetettu oppiainekohtaiset osaamistavoitteet 6. luokan lopulle vastaamaan hyvää osaamista ja arvosanaa 8. Oppiainekohtaisten tavoitteiden lisäksi on asetettu laaja-alaiset tavoitteet. Ohjelmointiin liittyviä tavoitteita esiintyy sekä vuosiluokkien 1-2 että 3-6 tavoitteissa matematiikan oppimistavoitteissa ja laaja-alaisissa tieto- ja viestintäteknologian osaamiseen (L5) liittyvissä tavoitteissa. Ohjelmointi mainitaan myös vuosiluokkien 3-6 käsityön tavoitteissa. (Opetushallitus 2014)

Ohjelmointi on sisällytetty osaksi matematiikan tavoitteiden sisältöaluetta "*S2 Ajattelun taidot*". Vuosiluokkien 1-2 osalta kyseisessä osiossa ohjeistetaan:

*"Tutustuminen ohjelmoinnin alkeisiin alkaa laatimalla vaiheittaisia toimintaohjeita, joita myös testataan."* (Opetushallitus 2014; 129).

Vuosiluokkien 3-6 osalta samaisessa osiossa ohjeistetaan:

*"Suunnitellaan ja toteutetaan ohjelmia graafisessa ohjelmointiympäristössä."* (Opetushallitus 2014; 235).

Lisäksi matematiikan opetuksen tavoitteissa on ohjeistus T14:

*"innostaa oppilasta laatimaan toimintaohjeita tietokoneohjelmina graafisessa ohjelmointiympäristössä."* (Opetushallitus 2014; 235).

Hyvän osaamisen eli arvosanan 8 saavuttamiseen matematiikassa 6. luokan päätteeksi edellytetään, että oppilas osaa luoda toimivan ohjelman graafisessa ohjelmointiympäristössä (Opetushallitus 2014).

Laaja-alaisen osaamisen tavoitteissa ohjelmointi mainitaan Tieto- ja viestintäteknologinen osaaminen (L5)- osiossa. Vuosiluokkien 1-2 laaja-alaisissa tavoitteissa mainitaan:



*“Oppilaat saavat ja jakavat keskenään kokemuksia digitaalisen median parissa työskentelystä sekä ikäkaudelle sopivasta ohjelmoinnista. Pelillisyyttä hyödynnetään oppimisen edistäjänä.”* (Opetushallitus 2014; 101)

Vuosiluokkien 3-6 osalta samainen osio sisältää seuraavan tavoitteen:

*“Ohjelmointia kokeillessaan oppilaat saavat kokemuksia siitä, miten teknologian toiminta riippuu ihmisen tekemistä ratkaisuista.”* (Opetushallitus 2014; 157).

Lisäksi vuosiluokkien 3-6 käsityön tavoitteissa mainitaan ohjelmoinnin harjoittelu osiossa *S3 Kokeilu*. Toimintoja kokeillaan robotiikassa ja automaatiassa. (Opetushallitus 2014; 239).

Perusopetuksen opetussuunnitelman 2014 perusteella ohjelmointi on noussut ja nousemassa olennaiseksi opetettavaksi aihe-alueeksi erityisesti matematiikan ajattelun taitojen osalta. Ajattelun taitojen tavoitteet tiivistyvät alakoulun osalta matematiikan hyvää osaamista vastaavaan tavoitteeseen: Oppia ohjelmoimaan jokin ohjelma graafisessa ohjelmointiympäristössä.

Opetussuunnitelman asettama haaste ohjelmoinnin opetuksen osalta on tavoitteiden epätarkkuus: Mitä ominaisuuksia oppilaan tekemän ohjelman tulee sisältää? Ohjelmoinnin opetusta koskevat kohdat perusopetuksen opetussuunnitelman perusteissa ovat hyvin yleisiä eivätkä sisällä tarkkaa oppimistavoitetta. Opetuksen järjestäminen jää opettajalle tulkinnanvaraiseksi ja antaa toisaalta vapauksia. Raja vedetään kuitenkin graafiseen ohjelmointiympäristöön. Riittää, että oppilas osaa ohjelmoida yhdessä graafisessa ohjelmointiympäristössä täyttääkseen hyvän osaamisen matematiikassa ohjelmoinnin osalta.

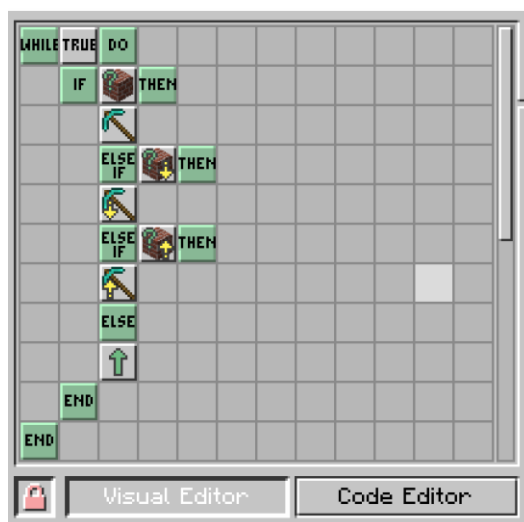
## **2.2 Graafinen ohjelmointiympäristö**

Ohjelmoinnin opettamisesta lapsille ei juurikaan didaktisia teoksia löydy, vaan kiinnostus on kohdistunut enemmänkin erilaisten lapsiystävällisten ohjelmointiympäristöjen käyttöön opetusvälineenä ja oppimisympäristönä. 2000-luvulla kirjoitetuissa ohjelmoinnin opettamiseen liittyvissä artikkeleissa käsitellään paljon lapsiystävällisiä ohjelmointiympäristöjä ja –kieliä sekä niiden käyttöä opetuksessa. Esimerkkejä tällaisista ohjelmointiympäristöistä ovat *Scratch*, *ToonTalk*, *LOGO* ja *Stargecast Creator* (Fessakis

et al., 2013). Tässä osiossa käsittelen graafista ohjelmointia ja graafisten ohjelmointiympäristöjen erityispiirteitä.

Graafista ohjelmointia ja yksinkertaisempia ohjelmointikieliä on kritisoitu niiden helppouden takia. Alkava ohjelmoija opetetaan tiettyyn ohjelmointikieleen ja kun vaikeampi kieli tulee vastaan, sitä on vaikea käyttää. Tässä vaiheessa tullaan kysymykseen, mitä ohjelmoinnin opetuksella halutaan opettaa? Aloittelevan ohjelmoijan oppimistavoitteena voidaan pitää ohjelmoinnin perusteiden ja algoritmisen ajattelun oppimista sekä antaa valmiudet tuleville opinnoille, eivätkä muut sisällöt saa vaikuttaa poissulkevasti tapaan saavuttaa nämä oppimistavoitteet (Mannila et al. 2007). Perusasioiden oppiminen asetetaan edelle ohjelmointikieltä ja ohjelmointiympäristöä valittaessa.

Graafinen ohjelmointiympäristö nousee uusimmassa perusopetuksen opetussuunnitelmassa tärkeään rooliin. Oppilaan tulisi oppia ohjelmoimaan nimenomaan graafisessa ohjelmointiympäristössä (Opetushallitus 2014). Tämä saattaa olla opetushallituksen tapa osoittaa opettajille lapsiystävällinen ohjelmointiympäristö. Olisi merkillistä, jos opetushallitus ajattelisi graafisen ohjelmointiympäristön olevan jollain muulla tavalla perinteistä ohjelmointiympäristöä parempi ohjelmointiympäristö.



*Kuvassa vasemmalla kuvakkeilla tehty ohjelma ja oikealla sama ohjelma tekstimuodossa. Koodi tehty Minecraftin ComputerCraftEdu-modissa.*

### 2.3 Algoritminen ajattelu

Käsite “Computational thinking” on ohjelmoinnin oppimista käsittelevissä tutkimuksissa vahvasti esillä. Käsite tuli esille ensimmäisen kerran Seymour Papertin tutkimuksissa, joissa hän käytti käsitettä kuvaamaan ajattelutapaa, jota tarvitaan ohjelmoinnin ongelmien ja tehtävien ratkaisemisessa (Papert 1996). Virallista käännöstä käsitteelle ei ole, mutta “laskennallinen ajattelu” ja “algoritminen ajattelu” ovat kohtuullisen toimivia suoria käännöksiä. Tässä tutkielmassa käytän käännöstä “algoritminen ajattelu”.

Algoritminen ajattelu on tärkeä taito, vaikkei koskaan laitteita tai ohjelmia ohjelmoisikaan. Algoritminen ajattelu on äärimmäisen loogista, mutta abstraktia ajattelua. Tämän kaltainen ajattelu auttaa hyvin suurtenkin ongelmien ratkaisemisessa antamalla työkaluja ongelman luokitteluun ja pilkkomiseen. Näin monimutkaiseenkin ongelmaan voidaan löytää yksinkertainen ratkaisu. Samanlaisia elementtejä on matemaattisessa ongelmanratkaisussa ja algoritminen ajattelu onkin olennainen osa matemaattista ajattelua. Tietokone ei osaa ajatella kuten ihminen, eikä algoritmisessa ajattelussa ole tarkoitus ajatella kuten tietokone. Ihmisen älykkyys tekee tietokoneesta mielenkiintoisen ja älykkään. Algoritminen ajattelu on tavanomainen ajattelutapa tietojenkäsittelytieteissä. Ohjelmointi ja matemaattinen ongelmanratkaisu ovat luontevia tapoja harjaantua algoritmisessa ajattelussa. (Wings 2006)

### **3 Ohjelmoinnin oppimisen haasteita**

Oppimisympäristön luomisessa ja ohjelmoinnin opetuksessa on omat haasteensa. Ohjelmointiympäristö voi toimia ohjelmoinnin opetuksessa oppimisympäristönä, jos se pystyy vastaamaan oppimisympäristölle asetettuihin vaatimuksiin ja ohjelmoinnin oppimiselle asetettuihin haasteisiin. Haasteet 1 ja 2 liittyvät suoraan ohjelmoinnin oppimiseen ja haasteet 3 ja 4 oppimisympäristölle asetettuihin vaatimuksiin.

#### **3.1 Haaste 1: Irtonaisuus arjesta ja fyysisen laitteen ongelma**

J. Rogalski ja R. Samurçay [1990] tutkivat 80- ja 90-luvulla ohjelmointia ja ohjelmoinnin oppimista. Teoksessaan *“Psychology of programming”* Rogalski ja Samurçay ottivat kantaa ohjelmoinnin oppimisen kognitiivisiin haasteisiin.

Ohjelmoinnin oppimisessa ensimmäinen ongelma on jokapäiväisten ohjelmoinnin kaltaisten aktiviteettien puute. Arkisissa askareissa ei suoranaisesti harrasteta algoritmista ajattelua. Toinen ongelma on se, että ohjelmointi tapahtuu fyysisessä laitteessa, jonka toiminta ei oppijalle ole alkujaan selvä. (Rogalski & Samurçay 1990)

Tietokoneet ovat kehittyneet 1990-luvulta tähän päivään huomattavasti ja tietokoneet, esimerkiksi älypuhelimien ja tablettitietokoneiden muodossa, ovat valtaosan käytössä päivittäin. Tilastokeskuksen vuoden 2015 lopulla julkaistun tilaston mukaan 16-74 -vuotiaista suomalaisista 74% käytti internetiä päivittäin ja 69% omisti älypuhelimien (Tilastokeskus 2015). Muutoksen myötä Rogalskin ja Samurçayn mainitsemista haasteista ensimmäinen ei enää nykyään ole yksiselitteinen, mutta käyttöjärjestelmien käyttäjäystävällisyys saattaa edelleen pidättää käyttäjän ajatukset irrallaan ohjelmoinnista laitetta käyttäessään. Toisenkin haasteista säilyy tähän päivään, sillä laitteet kehittyvät ja monimutkaistuvat, joten fyysisen laitteen toiminta ei ole helposti opittavissa. Positiivista on kuitenkin se, että ohjelmointikielet eivät ole ainakaan monimutkaistuneet.

#### **3.2 Haaste 2: Ratkaisun esittäminen tietokoneen ymmärtämässä muodossa**

Useassa tutkimuksessa puhutaan aloittelevien ohjelmoijien vaikeuksista ja vertaillaan aloittelijan ja asiantuntijan osaamista ohjelmoida. Leon E. Winslow tutkii raportissaan *“Programming Pedagogy -- A Psychological Overview”* taitotason merkitystä ohjelmoinnissa tarkastelemalla ohjelmointia ongelmanratkaisuna. Hän kertoo

ongelmanratkaisutaidon olevan tärkeä taitotason indikaattori. Ongelmanratkaisutaidon on kuitenkin huomattu ulottuvan yli aihealueiden, erityisesti matemaattis-luonnontieteellisissä aihepiireissä, riippumatta henkilön kokemuksesta aiheen parissa. Uudelle osaamisen tasolle päästäänkin ratkaisemalla ongelmia. Ohjelmoinnissa esiintyvän ongelman ratkaisun Winslow jakaa neljään vaiheeseen:

1. Ongelman ymmärtäminen
2. Ratkaisutavan suunnittelu
  - a) jossain muodossa ja
  - b) tietokoneelle sopivassa muodossa
3. Ratkaisun kääntäminen ohjelmointikielelle
4. Testaus ja korjaukset

Aloittelevan ohjelmoinnin oppijan vaikeudet johtuvat usein heikoista ongelmanratkaisutaidoista. Vaikka ongelmanratkaisutaidot olisivat kohtuulliset, aloittelijoilla on myös ongelmia siirryttäessä ongelmanratkaisun vaiheesta 2a vaiheeseen 2b. (Winslow 1996) Kun aloitteleva ohjelmoija ymmärtää ongelman, hän usein myös osaa suunnitella ratkaisun jossain muodossa, mutta ratkaisun siirtäminen tietokoneen ymmärtämään muotoon on haasteellista.

Edellä kuvattu malli on muunnos George Pólyan ongelmanratkaisun mallista: 1. Understanding the problem, 2. Devising a plan, 3. Carrying out the plan, 4. Looking back (Pólya 1945).

### **3.3 Haaste 3: Häiriötekijöiden minimointi ja helppokäyttöisyys**

Donald Normanin mukaan oppimisympäristölle voidaan asettaa seitsemän vaatimusta, jotka sen tulee toteuttaa:

1. Tarjota runsaasti vuorovaikutusta ja palautetta oppijan ja oppimisympäristön välillä
2. Sisältää tarkkoja tavoitteita ja vakiintuneita toimintatapoja
3. Motivoida
4. Tarjota jatkuvaa haastetta oppijalle, mutta ei missään vaiheessa turhauttaa vaikeudellaan tai tylsistyttää helppoudellaan.
5. Antaa sitoutumisen tunnetta toiminnalle: Oppija tuntee yhteyttä oppimisympäristöön
6. Tarjoata sopivat työvälineet niin, että ne auttavat tehtävässä toimimatta häiriötekijöinä

7. Ei sisällä häiriötekijöitä, jotka voivat vaikuttaa haitallisesti yksilöllisen kokemuksen.  
(Norman 1993)

Jotta oppiminen olisi tehokasta ja mahdollisimman luonnollista, oppimisympäristöllä täytyisi pystyä vastaamaan edellä mainittuihin vaatimuksiin. Virtuaalisissa oppimisympäristöissä tämä on suuri haaste erityisesti häiriötekijöiden osalta. Pelit, joita ei varsinaisesti ole tarkoitettu opetuskäyttöön, sisältävät runsaasti häiriötekijöitä. Pelin käyttö oppimisympäristönä vaatiikin runsaasti rajoittamista, jotta oppija keskittyisi tehtävän suorittamiseen. Voi myös olla, että oppimisympäristö ei ole suurin häiriötekijä, vaan laitteen, vaikkapa tablet-tietokoneen, muut ohjelmat houkuttelevat enemmän. Molempia tilanteita voi verrata liikuntasaliin, jossa yritetään opettaa telinevoimistelua salibandyn harrastajille, mutta lattialle on jätetty lojumaan mailoja ja palloja.

Itse pelin käyttäminen ei saa myöskään olla liian haastavaa, että ohjelmoinnin oppimiseen päästään esteettä. Näin itse oppimisympäristö toimii oppimisen esteenä. Vertauksena voidaan esittää tilanne, jossa tarkoituksena on oppia salibandyn pelaamista suoraan pelaamalla, vaikkei oppilas osaa vielä pitää edes mailaa kädessään oikein.

### **3.4 Haaste 4: Motivointi**

Peruskoulun ohjelmoinnin opetuksesta luovuttiin lähes kokonaan 90-luvulla. Ohjelmoinnin oppiminen ja opettaminen tuntui turhauttavalta ja vaikealta. (Saarikoski 2006)

Motivaatio on yksi olennaisimmista tekijöistä oppimisessa. Decin ja Ryanin (1985) itsemääräämisteorian mukaan motivaatio voidaan jakaa sisäiseen ja ulkoiseen motivaatioon. Henkilön ulkoinen motivaatio kumpuaa sosiaalisesta paineesta. Sisäinen motivaatio taas johtuu henkilön omasta kiinnostuksesta. Ulkoisen ja sisäisen motivaatio siis kuvaavat, miksi henkilö on motivoitunut tehtävän suorittamiseen. Juuri sisäinen motivaatio on merkittävä oppimisen kannalta. Opetuksen järjestämisessä tulisi ehdottomasti ajatella, kuinka oppilas saadaan kiinnostumaan opetettavasta asiasta ja kuinka oppimisesta saadaan palkitsevaa. (Deci & Ryan 1985)

Tämä haaste liittyy suoraan Normanin oppimisympäristön 3. ja 4. vaatimukseen, sekä hieman myös 1. vaatimukseen. Tehtävien suorittaminen ei tunnu palkitsevalta, vaan liian vaikealta tai tylsältä, eikä oppilas kykene niistä suoriutumaan kunnialla. Näin oppilaan sisäinen motivaatio tehtävän suorittamista kohtaan jää vähäiseksi.

## 4 Minecraft oppimisympäristönä ohjelmoinnin opetuksessa

### 4.1 Mikä on Minecraft?

Minecraft on alunperin Markus Perssonin luoma videopeli, jonka kehittämistä varten hän perusti myöhemmin yrityksen Mojang AB (Mojang 2011). Vuonna 2014 Microsoft osti pelin oikeudet Mojang AB:lta 2.4 miljardilla dollarilla (Ovide & Rusli 2014). Minecraft on moninpeli-sandbox peli, jossa maailma koostuu erilaisista kuutioista, joita voi rikkoa ja asettaa uudelleen (Canossa et al. 2013). PC-version lisäksi pelistä on julkaistu omat versiot myynnissä oleville Playstation- ja XBOX -konsoleille, Nintendo Wii U- konsolille sekä tablettiversio Minecraft: Pocket edition (Mojang 2016). Pelin perinteisin tavoite on selviytyä vaarallisessa maailmassa hankkimalla resursseja sekä hyödyntää niitä selviytymistaitoja ja luovuutta käyttäen (Duncan 2011). Minecraft on kuitenkin kasvanut käsittämättömän monimuotoiseksi ja mahdollisuuksille avoimeksi pelimaailmassa päivitysten ja modauksen, eli pelin ominaisuuksia ja mahdollisuuksia modifioivien lisäosien, ansiosta. Selviytymisen ja rakentelun lisäksi pelin sisällä voi rakentaa hyvinkin monimutkaisia laitteita ja laitteistoja, esimerkiksi toimivan laskimen (Nebel et. al. 2016).



*Minecraft maailma rakentuu erilaisista kuutioista, joilla on erilaisia ominaisuuksia*

## **4.2 Minecraft opetuskäytössä ja oppimisympäristönä**

Peli on opetuskäyttöön soveltuva sellaisenaan, mutta Mojangin tukema, alkujaan suomalainen yritys TeacherGaming LLC on tehnyt Minecraftista erillisen opetuskäyttöön modifioidun MinecraftEdu-version (Nebel et. al. 2016). Microsoft osti vuoden 2016 alussa MinecraftEdu:n TeacherGaming LLC:ltä (Kauppalehti 2016) ja on alkanut kehittää uutta Minecraft: Education edition- versiota pelistä (Mojang 2016).

Minecraft voidaan käsittää virtuaalisena oppimisympäristönä. Minecraftin pelimaailmassa voidaan opettaa ja oppia eri oppiaineiden sisältöä. Matematiikan opetusta on testattu Minecraft maailmassa hyödyntäen pelin rakentamisominaisuuksia muotoja ja erilaisia rakennelmia tekemällä (Bos et. al. 2014). Lisäksi ainakin saksan kielen (Pihkala-Posti 2015) ja englannin kielen (Uusi-Mäkelä 2015) opetusta on kokeiltu Minecraftissa. Biologian opetusta on kokeiltu ohjaamalla oppilasta toimimaan ikään kuin pienoishahmona maailmassa, joka on jäljitelmä ihmisruumiista (Short 2012). Erilaista oppiaine sisältöä opetukselliseen tarkoitukseen luodaan jatkuvasti (Short 2012; Brand & Kinash 2013).

## **4.3 Minecraft ohjelmoinnin opetuksessa ja algoritmisen ajattelun edistäjänä**

Viime vuosina pelien käyttö oppimisympäristönä on alkanut kiinnostaa tutkijoita ja opettajia oppilaan motivoimisen ja oppimisen itsesäätelyn näkökulmasta (Huang 2010, Wilkinson et. al. 2013, Schifter et al. 2013, Canossa et. al. 2013.). Jo 1990-luvulla Robert Moser käsitteli ohjelmoinnin oppimisen haasteellisuutta ja ehdotti yhdeksi ratkaisuksi ohjelmoinnin opettamista fantasia-seikkailu pelissä yhtenä tärkeänä perusteluna oppilaan oma motivoituminen aiheeseen pelin kautta (Moser 1997). Yllä mainittujen tutkimusten lisäksi useissa aiheeseen liittyvissä tutkimuksissa ja artikkeleissa (Repenning et al. 2014; Zorn et. al. 2013; Bayliss 2012; Pellas 2014) pidetään motivointia yhtenä tärkeimmistä syistä pelien, kuten Minecraftin, käyttöön opetustarkoituksessa.

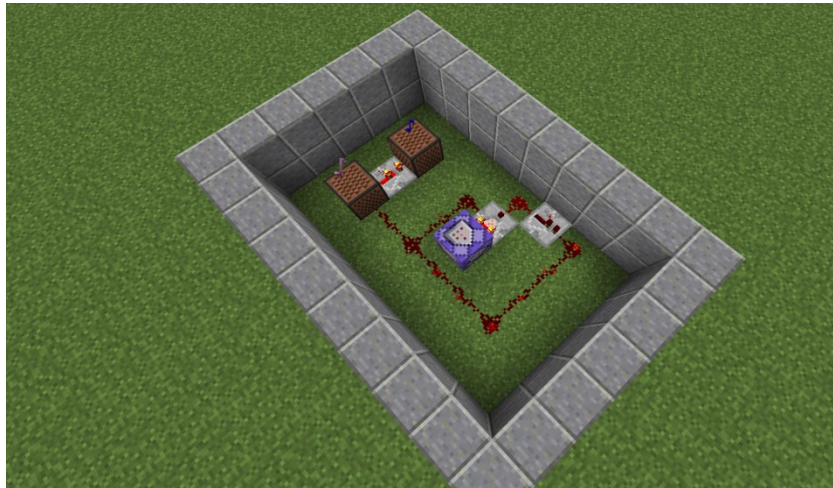
### **4.3.1 Redstone eli punakivi**

Punakivi on Minecraftissa sähkövirran lähde, jolla on mahdollista luoda erilaisia yksinkertaisia tai hyvinkin monimutkaisia laitteita yhdistelemällä eri punakivikomponentteja (Brand & Kinash 2013). Punakiven avulla voidaan rakentaa logiikkapiirejä ja digitaalipiirejä, jotka toimivat samankaltaisesti kuin oikean maailman vastaavat piirit (Short 2012). Lisäksi Minecraftiin on uusimmissa päivityksissä kehitetty



entistä parempia komentokuutioita, joiden avulla pelaaja voi antaa palvelimelle komentoja (Mojang 2016). Komentokuutioita ja punakivikomponentteja yhdistelemällä laitteistoista voidaan tehdä entistä moniulotteisempia (Mojang 2016).

Haasteeseen 1 voidaan osalti vastata punakivilaitteiden rakentamisella: Minecraftissa on mahdollista rakentaa digitaalipiirejä ja toimivia laitteistoja. Ohjattu punakivilaitteistojen rakentaminen voi olla yksi ratkaisu ainakin Rogalskin ja Samurçayn mainitsemaan fyysisen laitteen toimintalogiikan haasteeseen. Tämä ei kuitenkaan tee algoritmista ajattelusta jokapäiväistä, ellei harrasta punakivellä rakentelua joka päivä. Toisaalta tämän kaltainen simulaatio voi auttaa oppijan itseohjautumista aiheen piirissä (Jiau et al. 2009). Näin punakiven käyttö voi innostaa tutkimaan maailmaa ja teknologian toimintaa uudessa valossa.



*Laite, jossa on yhdistetty komentokuutio ja punakivikomponentteja. Kun pelaaja tulee komentokuution lähelle, laite aktivoituu ja aiheuttaa hälytyksen.*

#### 4.3.2 Ohjelmoitavat kilpikonnat - ComputerCraft ja ComputercraftEdu

ComputerCraft on Daniel “dan200” Ratcliffen luoma Minecraft modi, jossa on mahdollisuus ohjelmoida kilpikonnaksi kutsuttuja robotteja LUA-ohjelmointikielellä (dan200 2013). Kyseisestä modista on tehty opetuskäyttöön suunnattu versio ComputerCraftEdu, jossa merkittävänä erona on mahdollisuus ohjelmoida graafisesti selkeitä kuvakkeita käyttämällä (dan200 2013). Edu-versiossa pelaaja voi liikutella kilpikonnaa, tuhota kuutioita ja rakentaa kuutioilla rakennelmia ilman erityistä tietämystä LUA- kielestä tai ohjelmoinnista (Nebel et al. 2015).

Visuaalisuus ja selkeys voivat auttaa alkavaa ohjelmoijaa asettamaan suunniteltu ratkaisu tietokoneen ymmärtämään muotoon. Graafisille ohjelmointiympäristöille ominaisen välittömän palautteen on huomattu tukevan varsinkin esi- ja alkuopetuksessa olevia lapsia ohjelmoinnissa esiintyvien ongelmien ratkaisemisessa (Fessakis et. al. 2012).

#### 4.3.3 Muita tapoja opettaa ohjelmointia Minecraftissa

Minecraftin modaus on yksi tapa opettaa ohjelmointia. Minecraft on ohjelmoitu Java-ohjelmointikielellä, joten modifioimalla Minecraftin Java-koodia voidaan pelin sääntöjä ja ominaisuuksia muuttaa. Tätä kutsutaan Minecraftin “modaukseksi”. Modaamiseen tarvitaan ohjelmointityökalu, joka tukee Java Development Kit- ohjelmaa. (Cadenhead 2014; Gupta & Gupta 2015)

LearnToMod on ohjelmoinnin oppimista varten on tehty modi, jonka avulla Minecraftia voi modata graafisilla kuvakkeilla. Opetuksellinen apuohjelma opettaa ohjelmointia samalla kun oppilas pelaa Minecraftia. Kyseistä modia on kokeiltu opetuskäytössä ja sen käytön mahdollisuuksiin uskotaan. (Risberg 2015)

Christopher Zorn [2012] tutkimustiimeineen kehitti oman CodeBlocks-ohjelmointikielen ohjelmoinnin opetusta käsittelevää tutkimusta varten. Kyseinen ohjelmointikieli toimii CodeBlocks-modilla, jossa ohjelmointikoodin tekeminen tapahtuu asettamalla kuutioita riviin kuvaamaan haluttua koodia. Ohjelmoinnin kohteena on robottikuutio, joka suorittaa pelaajan rakentaman koodin. Modin tekijät saivat inspiraation omaan CodeBlocks-ohjelmointikieleen Scratch- ja StarLogoBlocks-ohjelmointiympäristöistä. (Zorn et. al. 2013)

## 5 Aiheesta tehty aiempi tutkimus

Aiempaa tutkimusta aiheesta on vähän, sillä Minecraft ja sen suosio ovat aika tuoreita ilmiöitä. Tässä osiossa otan esille muutamia tutkimuksia, joissa on tutkittu ohjelmoinnin opetusta Minecraftissa. Yksi tutkimuksista on vielä “Work in Progress” tilassa, mutta tutkimus on jo suoritettu ja tulokset on analysoitu.

### 5.1 Tutkimus A: ComputerCraft - innostava ohjelmointiympäristö

Tässä tutkimuksessa Brett Wilkinson, Neville Williams ja Patrick Armstrong tutkivat, voiko Minecraftin käyttö ohjelmoinnin opetuksessa helpottaa ohjelmoinnin opetusta lapsille ja motivoida oppilaita ohjelmoinnin oppimiseen tarjoamalla tutun, turvallisen ja motivoivan ohjelmointiympäristön. Tutkijoiden mielestä ohjelmoinnin opettaminen on vaikeaa, erityisesti kun kohteena ovat lapset. Wilkinson ja kumppanit vertaavat ohjelmoinnin oppimista vieraan kielen oppimiseen. Videopelin käyttö ohjelmointiympäristönä nähdään mahdollisena ratkaisuna ohjelmoinnin opettamisen haasteeseisiin. (Wilkinson el. al. 2013)

Tutkimuksessa opetettiin ohjelmointia Minecraftissa LUA-ohjelmointikielillä ComputerCraft-modia käyttäen. Ohjelmointi tapahtui tekstipohjaisesti, mutta palaute ohjelmoinnin onnistumisesta on kyseisessä modissa graafista ja välitöntä. Tutkimus toteutettiin osana ohjelmointityöpajaa, johon osallistui yli 500 8-16 vuotiasta lasta. Lapset työskentelivät omien toivomustensa mukaan yksin tai pareittain, yksinpelissä tai monipelissä. Oppilasta ohjattiin ensin ymmärtämään sisältö demonstraatioilla, keskustelulla ja käytännön sovelluksilla. Kun sisältö ymmärrettiin, oppilasta ohjattiin analysoimaan ratkaisemaan ohjelmoinnilla ratkaistavaa ongelmaa. Lopulta oppilas sai soveltaa osaamistaan omilla haluamillaan tavoilla. Minecraftin ja ComputerCraftin toimivuutta pedagogisena välineenä mitattiin muokatulla versiolla Bloomin taksonomiasta. (Wilkinson el. al. 2013)

Osalta (14%) oppilaista kerättiin palautetta työpajan toiminnasta haastatteluilla. Suurin osa oli tyytyväisiä työpajaan. 30% haastateltavista oli ohjelmoinut aikaisemmin, suurin osa graafisilla ohjelmointikielillä, ei tekstipohjaisilla. Ne oppilaat, joiden mielestä tietojenkäsittely tieteet eivät olleet kiinnostavia, eivät ennen työpajaan osallistumista tienneet tietojenkäsittelyn sisältävän juuri kokemaansa luovuutta ja hauskuutta. 69%

kyseisistä oppilaista kertoi muuttaneen mielensä työpajan ansiosta. 88% haastateltavista osasi selittää työpajassa tehtyyn abstraktiin ongelmaan tyydyttävän ratkaisun. Uudelleen työpajaan osallistuneita oppilaita haastateltiin vielä erikseen. 71% mielestä ohjelmointiympäristöä oli tässä vaiheessa helppo käyttää. 91% haastateltavista kertoi nauttivansa kokemuksesta. 57% haastateltavista sanoi jatkaneensa ohjelmointia kotona ja pyrkineensä oppia lisää aiheesta. (Wilkinson et al. 2013)

## **5.2 Tutkimus B: Ohjelmoinnin oppimisesta hauskaa ja helppoa CodeBlocksin avulla**

Tässä tutkimuksessa Christopher Zorn, Chadwick Wingrave, Emiko Charbonneau ja Joseph J. LaViola Jr. kokeilivat kehittämänsä CodeBlocks- ohjelmointikieltä Minecraftissa opettaakseen ohjelmointia. Tutkimuksen tavoitteena oli tutkia, voiko CodeBlocksin käytöllä ohjelmointiympäristönä lisätä kiinnostusta ohjelmointia kohtaan, voiko aloittava ohjelmoija ratkaista pulmia CodeBlocksilla ja voiko CodeBlocksin käytöllä kehittyä ohjelmoinnin osaamisessa. (Zorn et al. 2013)

Tutkimuksessa CodeBlocksia kokeiltiin kahdessa ryhmässä. Toisessa käytettiin tekstipohjaista ohjelmointia ja toisessa kuutioilla tehtävää graafisempaa ohjelmointia. Osallistujat (n=30) olivat 18-51 vuotiaita psykologian opiskelijoita, joista 15 oli miehiä ja 15 naisia. Ensin osallistujat suorittivat esitietokysely, jossa vastaukset kerättiin 7 pisteen Likert skaalalla. Osallistujilta kysyttiin heidän kiinnostusta ja asenteita ohjelmointia ja robotteja kohtaan. Lisäksi suoritettiin kartoitus osallistujien oppimistavoista. Seuraavaksi osallistujat opastettiin CodeBlocksin käyttöön, jonka jälkeen heidän tuli suorittaa neljä yksinkertaista ongelmanratkaisua vaativaa harjoitusta. Viimeisenä haasteena osallistujat saivat valita neljästä isommasta pulmasta yhden ratkaistavakseen itsenäisesti. Lopuksi suoritettiin vielä sama kysely kuin aluksi. (Zorn et al. 2013)

Merkittävä havainto tutkimustuloksissa oli kasvanut kiinnostus ohjelmointia kohtaan. Sekä tekstipohjaisesti ohjelmoineet että kuutioilla ohjelmoineet ilmaisivat nauttivansa enemmän ohjelmoinnista kokeilun jälkeen. Kiinnostus ohjelmointia kohtaan kasvoi myös huomattavasti molemmissa ryhmissä. Ohjelmointia ei myöskään nähty enää niin vaikeana tai tarpeettomana kuin ennen tutkimukseen osallistumista. Huomattavaa eroa ryhmien välillä ei ollut. Zorn ja kumppanit kokeilivat CodeBlocksin käyttöä myös STEM kesäleirillä, johon osallistui useita ryhmiä yläkouluikäisiä lapsia. Leirille osallistuneista suurin osa oli kiinnostunut kokeilemaan CodeBlocksia uudestaan. Zorn ja kumppanit

arvelevat CodeBlocksin toimivan hyvin Minecraft pelaajien ja ohjelmointia harrastamattomien kiinnostuksen herättäjänä ohjelmointia kohtaan. (Zorn et. al. 2013)

### **5.3 Tutkimus C: Aiheesta kiinnostuminen Minecraftin avulla**

Veronica Cateté, Kathleen Wassell ja Tiffany Barnes tutkivat tässä tutkimuksessa SPARCS-ohjelman (Students in Programming, Robotics, and Computer Science) toimivuutta osana koulun jälkeistä opetuksellista kerhotoimintaa. Tavoitteena oli löytää ja tarkastella niitä tekijöitä, jotka houkuttelevat lapsia tietojenkäsittelytieteiden ääreen. SPARCS-ohjelmaan kuului unplugged-harjoituksia, ohjelmointia ALICE-ohjelmointiympäristössä, verkkosivujen suunnittelua, tietoturvallisuus opetusta, johdatus mikropiirien toimintaan, mobiilisovellusten kehittämistä, robotiikkaa ja GameMaker-ohjelman käyttöä pelien tekemisessä. Minecraft oli osana mikropiirien toiminnan opetuksen toiminnallista osuutta. Oppitunnilla tehtiin logiikkapiirejä, joilla muun muassa avattiin ja lukittiin ovia Minecraftissa. (Cateté et. al. 2014)

Cateté tutkimustiimeineen oli aikaisemmassa tutkimuksessaan todennut, että mitä hausempaa kurssilla on, sitä todennäköisemmin lapsi pysyy mukana kurssilla. Tämän vuoksi tutkimuksen aineisto hankittiin kyselyillä, joilla kartoitettiin kuinka paljon osallistuja nautti kurssilla mukana olemisesta. Tutkimustuloksista selvisi, että osallistujat nauttivat eniten viihdeteknologiaan liittyvistä harjoitteista, joissa tehtiin itse pelejä ja mobiilisovelluksia. Poikkeuksena oli mikropiirin opetus, joka nähtiin yhtä mukavana kuin mobiilisovellusten tekeminen. Satunnaisista haastatteluista ilmeni, että juuri Minecraftin käyttö oli suuri syy tähän poikkeukseen. (Cateté et. al. 2014)

### **5.4 Tutkimus D: Teksti- vai kuvakepohjainen ohjelmointi?**

Daisuke Saito, Hironori Washizaki ja Yoshiaki Fukazawa tutkivat tutkimuksessaan graafisen ja tekstipohjaisen ohjelmoinnin eroa ohjelmoinnin oppimisen näkökulmasta. Tutkimuksessa käytetään Minecraftin ComputerCraftEdu-modia, jossa kilpikonna voidaan ohjelmoida sekä graafisilla kuvakkeilla että tekstipohjaisesti. Osallistujat (n=16) jaettiin kahteen ryhmään, joista toisessa oli 11 graafisilla kuvakkeilla ohjelmoijaa ja toisessa 5 tekstipohjaisesti ohjelmoijaa. Opetus toteutettiin kolmen tunnin työpajana, jossa opeteltiin ComputerCraftEdun käyttö (Basic), Silmukka (Loop), ehtolause (If) ja lopulta kilpikonna ohjelmoitiin rakentamaan osallistujan etunimen ensimmäinen kirjain. (Saito et. al. 2015)

Tutkimuksen aineisto kerättiin Likert-skaalalla pisteytetyllä kyselyllä. Graafisilla kuvakkeilla ohjelmoineen ryhmän kiinnostus ohjelmointia kohtaan kasvoi huomattavasti kokeilun tuloksena. Myös kilpikonna-ohjelmoinnin hyöty ja halukkuus ohjelmoida kasvoivat huomattavasti. Tekstipohjaisesti ohjelmoineen ryhmän tulokset eivät olleet yhtä selkeät. Huomattavaa oli kuitenkin se, että kokeilun tuloksena osa ryhmästä näki kilpikonna-ohjelmoinnin helpompana, mutta osa näki sen myös tarpeettomampana kuin ennen kokeilua. (Saito et. al. 2015)

On valitettavaa, että osallistujamäärä tekstipohjaisessa ryhmässä oli niin pieni, ettei ryhmien välisiä tuloksia voida oikein verrata. Tutkimus on toistaiseksi *Work in Progress*-vaiheessa.

## 6 Johtopäätökset ja pohdinta: Kuinka Minecraft vastaa ohjelmoinnin oppimisen haasteisiin?

Tässä tutkielmassa käsittelin ohjelmoinnin oppimista ja Minecraftin käyttöä ohjelmointiympäristönä pääasiassa peruskoulun opetukseen liittyen. Ohjelmoinnin opetusta tarkastelemalla sain selville ne haasteet, joihin ohjelmoinnin opetuksessa tulisi pystyä vastaamaan. Teoriaa tarkastelemalla ohjelmoinnin oppimiseen ja oppimisympäristöön kohdistuviksi haasteiksi nousivat:

1. Irtonaisuus arjesta ja fyysisen laitteen ongelma
2. Ratkaisun esittäminen tietokoneen ymmärtämässä muodossa
3. Häiriötekijöiden minimointi ja helppokäyttöisyys
4. Motivointi

Rogalskin ja Samurçayn mainitsemaan ohjelmoinnin ja jokapäiväisten aktiviteettien irrallisuuteen, voidaan vaikuttaa “Unplugged”-leikeillä. Unplugged-toiminta on ikään kuin ohjelmointia ilman tietokoneita tai muuta elektroniikkaa. Siinä tulevat esille ohjelmoinnissa olennaiset taidot, kuten algoritminen ajattelu. Aktiviteetit sisältävät haasteita, joita lapset ratkaisevat itse. Aktiviteetit voivat olla esimerkiksi leikkejä, pelejä, pulmia tai temppeja, joihin ei tarvita mitään tiettyä laitetta. Unplugged- sisältö kasvaa jatkuvasti ja projektin kehittäjät toivovat sen kasvavan yhteistyön ja yhteisön kehittämänä maailmanlaajuisesti ilmiöksi. (Bell, et al. 2009) Unplugged toiminta koulussa opettaa lapsille uusia tapoja leikkiä ja ajatella leikkimistä. Unplugged-leikeistä saattaa tulla osa välitunnin itsenäisiä leikkejä lasten kesken. Näin ohjelmoinnista voi tulla yhdellä tavalla osa arkea. Unplugged- leikit eivät kuitenkaan suoranaisesti liity Minecraftiin. Minecraftin käyttö ei välttämättä ole luonnollinen tapa arkipäiväistä ohjelmointia. Fyysisen laitteen ongelmaan voidaan kuitenkin vaikuttaa opettamalla mikropiirien toimintaa Minecraftissa punakiveä käyttäen, kuten Tutkimus C:ssä (Cateté et. al. 2014). Haasteeseen 1 ei siis pelkästään Minecraftilla voida vastata.

Haasteessa 2 on kyse oppilaan ongelmanratkaisutaidoista ohjelmointiin liittyvissä ongelmissa ja tehtävissä. Haaste 2 voi olla osasy s siihen, että graafiset ohjelmointiympäristöt ovat tällä hetkellä suosiossa opetuskäytössä. Graafisen ohjelmointiympäristön välitön visuaalinen palaute voi auttaa aloittelevaa ohjelmoijaa ongelmien ratkaisussa (Fessakis et. al. 2012). Peruskomentojen ymmärtämisessä

kuvakkeista ei todennäköisesti ole huomattavaa etua tekstipohjaiseen ohjelmointiin verrattuna, jos oppilas osaa jo englantia. Ajallista etua voi olla siitä, että komennot voidaan siirtää valmiina koodiin, eikä tarvitse muuta kuin muokata komento haluttuun muotoon. Tällä tavalla toimii juuri kuvakepohjainen ohjelmointi. Ratkaisujen siirtäminen tietokoneen tai sovelluksen ymmärtämään muotoon vaatii kuitenkin peruskomentojen ymmärtämistä. Muuttujien ja peruskomentojen, kuten ehto- ja toistolauseen, opettamiseen tuleekin panostaa, ettei tietokoneen kieli muodostu haasteeksi.

Tutkimuksessa A algoritmien käyttö opittiin kokeilemalla niitä Minecraftin graafisessa ComputerCraft-ohjelmointiympäristössä. Tutkimukseen osallistuneita pyydettiin työpajan jälkeen selittämään omin sanoin juuri kyseisiä algoritmeja. Suurin osa osallistuneista kykeni hyväksyttävästi selittämään näiden algoritmien käyttöperiaatteen. (Wilkinson et al. 2013)

Minecraftissa voidaan siis oppia ohjelmointia niin, ettei tietokoneen kieli koidu ylitsepääsemättömäksi esteeksi ongelmanratkaisussa. Haasteeseen 2 täytynee perehtyä tarkemmin jatkotutkimuksessa.

Kolmas haaste on tehdä oppimisympäristöstä helppokäyttöinen ja minimoida häiriötekijät. Virtuaalisessa oppimisympäristössä tämä tarkoittaa sitä, että oppija saa selkeän opastuksen siitä, miten peliä pelataan ja mitä hänen halutaan tekevän. Minecraftissa häiriötekijöiden minimointi toteutuu parhaiten toimintaa rajoittavilla modeilla. Esimerkiksi MinecraftEdu:n ajatus oli lähinnä helpottaa hallitsemaan peliä ja pelaajia, että opettaminen olisi vaivattomampaa ja häiriötöntä. Osaava Minecraftin käyttäjä pystyy kuitenkin hoitamaan samoja asioita Minecraftin alkuperäisessä versiossa. Nähtäväksi jää, kuinka Minecraft: Education edition vastaa oppimisympäristölle asetettuihin vaatimuksiin.

Jo vuonna 1998 Rob Houser ja Scott DeLoach ottivat artikkelissaan *Learning from games: Seven principles of effective design* (1998) kantaa hyvin toimivan pelin ominaisuuksiin. Heidän mielestään Normanin seitsemän vaatimusta oppimisympäristölle osoittavat, että pelaaminen on olennaisesti yhteydessä oppimiseen. He eivät puhu suoranaisesti peleistä opetustarkoituksessa vaan kuinka hyvin suunnitellusta pelistä voidaan ottaa mallia kaupallisten tietokoneohjelmien suunnittelussa. (Houser & DeLoach 1998)



Tutkimuksessaan Houser ja DeLoach pyrkivät tuomaan esille seitsemän periaatetta, jotka pätevät hyvin suunnitelluissa peleissä. Lyhyesti nuo periaatteet ovat:

1. Käyttäjän tulisi oppia ohjelman käyttö nopeasti ja vaivattomasti
2. Käyttäjän tulisi aina tietää mitä hänen tulee tehdä onnistuakseen
3. Käyttäjälle tulisi antaa vain juuri tarpeeksi neuvoja, että hän kykenee aloittamaan ja jatkamaan toimintaa
4. Tehtävän suorittamista ei saisi häiritä fyysisesti eikä psykologisesti itse ohjelman sisällä
5. Käyttäjälle tulisi antaa käytännöllisiä vinkkejä tehtävän suorittamiseen liittyen, joita hän voi hyödyntää tai olla hyödyntämättä
6. Käyttäjälle tulisi antaa välittömästi onnistumisen kokemuksia ohjelman käytössä. Vähitellen tulisi myös johdattaa käyttäjää tutustumaan ohjelman monimutkaisempiin osioihin.
7. Käyttäjän tulisi olla aina tietoinen siitä, kuinka tehokkaasti hänen toimintansa auttaa kohti tavoitetta. (Houser & DeLoach 1998)

Minecraftiin luotavaa virtuaalista oppimisympäristöä suunniteltaessa Houserin ja DeLoachin mainitsevat periaatteet sattavat osoittautua elintärkeiksi. Hyvin toteutetun pelin ominaisuuksilla pelaaja saadaan päättäväisesti pyrkimään kohti tavoitetta. Jos tavoitteeksi asetetaankin ongelma tai tehtävä, jonka suorittaminen edellyttää oppimistavoitteisiin kuuluvaa sisältöä, peli toimii tehokkaasti oppimisympäristönä. Koska kaupalliset ohjelmat voivat toimia näiltä osin pelin tavoin (Houser & DeLoach 1998), niin miksei samaa ajatusta voisi soveltaa opetusohjelmissa?

Tarkastelemissani aikaisemmissa tutkimuksissa Minecraft toimii ohjelmoinnin oppimisessa erityisesti innostuksen ja kiinnostuksen herättäjänä. *Tutkimus A:ssa* (Wilkinson et al. 2013) huomattiin, että 57% tutkimukseen osallistuneista kokeili ohjelmointia myös kotona vapaaehtoisesti. *Tutkimus B:ssä* (Zorn et al. 2013) saatiin vähän samankaltaisia tuloksia, nimittäin suurin osa osallistuneista oli valmis kokeilemaan tutkimuksessa käytettyä CodeBlocks-ohjelmointia uudelleen. *Tutkimus C:ssä* (Cateté et al. 2014) todettiin Minecraftin olevan innostava tekijä mikropiirien toiminnan oppimisessa. Myös *Tutkimus D:ssä* (Saito et al. 2015) havaittiin selkeitä viitteitä lisääntyneestä kiinnostuksesta ohjelmointia kohtaan. Nämä tulokset ovat merkkejä Decin ja Ryanin (1985)

itseääräämisteoriassa mainitusta sisäisestä motivaatiosta ohjelmointia kohtaan. Minecraftin modatulla käytöllä voidaan siis vastata haasteeseen 4.

Aikaisempien tutkimusten perusteella Minecraftin potentiaali ohjelmoinnin opetuksessa piilee sen motivoivassa ja innostavassa vaikutuksessa. Jos alkava ohjelmoija saadaan innostumaan ohjelmoinnista, ja näkemään ohjelmointi helppona, ollaan siinä vaiheessa ratkaistu ne ongelmat, joiden myötä ohjelmointi väheni minimiin peruskoulun opetuksessa. Todettakoon kuitenkin, että vaikka Minecraftissa voitaisiin moitteettomasti opettaa ohjelmointia, sen käyttö vaatii opettajalta paljon valmistelua ja tietotaitoa kyseisen pelin pelaamisessa. Todennäköinen jatkotutkimukseni onkin luoda opettajalle helppokäyttöinen Minecraft-maailma ohjelmoinnin opetukseen peruskoulussa ja kokeilla, vastaako se ohjelmoinnin opetuksen haasteisiin ja edistääkö se ohjelmoinnin oppimista. Oma kokemukseni on, että avoimen pelimaailman käyttö oppimisympäristönä vaatii hyvin vahvaa toiminnan kontrollointia niin, että toiminta vie kohti oppimistavoitteita. Vastaava ongelma on esillä mainitsemissani ohjelmoinnin opetuksen haasteissa. Tämä onkin yksi olennaisimmista haasteista jatkotutkimusta ajatellen.

## Lähdeluettelo:

- Opetushallitus. (2014). Perusopetuksen opetussuunnitelman perusteet. Tampere: Juvenes Print Suomen Yliopistopaino Oy.
- Opetushallitus. (2004). Perusopetuksen opetussuunnitelman perusteet. Vammala: Vammalan Kirjapaino Oy.
- Code.org (2015). *Hour of Code*. Noudettu sivustolta <https://hourofcode.com> 25.4.2016
- Saarikoski P. (2006). Koneen ja koulun ensikohtaaminen Suomalaisen atk-koulutuksen varhaisvaiheet peruskoulussa ja lukiossa. *Tekniikan Waiheita* 3/06.
- Papert, S. (1996). An exploration in the space of mathematics educations. *International Journal of Computers for Mathematical Learning*, 1(1), 95-123.
- Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33-35.
- Fessakis, G., Gouli, E., Mavroudi, E. (2013). Problem solving by 5–6 years old kindergarten children in a computer programming environment: A case study. *Computers & Education*, 63, 87–97. doi:10.1016/j.compedu.2012.11.016
- Suomen virallinen tilasto (SVT): Väestön tieto- ja viestintätekniiikan käyttö [verkkójulkaisu]. Taulukot: *Internetin käytön ja eräiden internetin käyttötapojen yleisyys 2015 ; Älypuhelin omassa käytössä 2015, %-osuus väestöstä*. ISSN=2341-8699. 2015. Helsinki: Tilastokeskus [viitattu: 20.4.2016].
- Byrne, P., & Lyons, G. (2001, June). The effect of student attributes on success in programming. In *ACM SIGCSE Bulletin* (Vol. 33, No. 3, pp. 49-52). ACM.
- Winslow, L. E. (1996). Programming pedagogy—a psychological overview. *ACM SIGCSE Bulletin*, 28(3), 17-22.
- Polya, G., Kaddouch, R., Renou, M., Comtois, M., & Dubois, M. J. (1957). How to solve it: a new aspect of mathematical method/G.
- Norman, D. A. (1993). *Things that make us smart: Defending human attributes in the age of the machine*. Basic Books.

- Deci, E. & Ryan, R. 1985. The general causality orientations scale: Self-determination in personality. *Journal of Research in Personality* 19 (2), 109–134.
- Bell, Tim, et al. "Computer science unplugged: School students doing real computing without computers." *The New Zealand Journal of Applied Computing and Information Technology* 13.1 (2009): 20-29.
- Mannila, L., Peltomäki, M., & Salakoski, T. (2006). What about a simple language? Analyzing the difficulties in learning to program. *Computer Science Education*, 16(3), 211-227. Turku: TUCS Dissertations.
- Persson, M., & Bergensten, J. (2016). Minecraft. *Computer software*. Stockholm, Sweden: Mojang AB. Noudettu sivustolta <http://minecraft.net>.
- Duncan, S. C. (2011). Minecraft, beyond construction and survival. *Well Played: a journal on video games, value and meaning*, 1(1), 1-22.
- Canossa, A., Martinez, J. B., & Togelius, J. (2013, August). Give me a reason to dig Minecraft and psychology of motivation. In *Computational Intelligence in Games (CIG), 2013 IEEE Conference on* (pp. 1-8). IEEE.
- Nebel, S., Schneider, S., & Rey, G. D. (2015). Mining learning and crafting scientific experiments: a literature review on the use of minecraft in education and research. *Educational Technology & Society*.
- Ovide, S., & Rusli, E. M. (2014). Microsoft Gets' Minecraft'-Not the Founders. *Viitattu 19.9.2014*.
- Kauppalehti 2016. Karjalainen: Parikymppinen suomalaisyrittäjä teki kaupat Microsoftin kanssa. Viitattu 12.5.2016. Noudettu sivustolta: <http://www.kauppalehti.fi/uutiset/karjalainen-parikymppinen-suomalaisyrittaja-teki-kaupat-microsoftin-kanssa/9NwBsm8e>
- Bergensten, J. (2016). Post: ANNOUNCING MINECRAFT: EDUCATION EDITION. *Stockholm, Sweden: Mojang AB*. Noudettu sivustolta <https://mojang.com>.
- Bos, B., Wilder, L., Cook, M., & O'Donnell, R. (2014). Learning mathematics through Minecraft. *Teaching Children Mathematics*, 21(1), 56-59.

- Pihkala-Posti, L. (2015). Pelillisyyttä kielenoppimiseen!
- Uusi-Mäkelä, M. (2015). Learning English in Minecraft: a case study on language competences and classroom practices.
- Short, D. (2012). Teaching scientific concepts using a virtual world—Minecraft. *Teaching Science-the Journal of the Australian Science Teachers Association*, 58(3), 55.
- Brand, J., & Kinash, S. (2013). Crafting minds in minecraft.
- Moser, R. (1997, June). A fantasy adventure game as a learning environment: why learning to program is so difficult and what can be done about it. In *ACM SIGCSE Bulletin* (Vol. 29, No. 3, pp. 114-116). ACM.
- Huang, W. H. (2011). Evaluating learners' motivational and cognitive processing in an online game-based learning environment. *Computers in Human Behavior*, 27(2), 694-704.
- Wilkinson, B., Williams, N., & Armstrong, P. Improving Student Understanding, Application and Synthesis of Computer Programming Concepts with Minecraft.
- Schifter, C. C., Cipollone, M., & Moffat, F. (2013). Piaget, Inhelder and Minecraft. *2013 celda*, 210.
- Repenning, A., Webb, D. C., Brand, C., Gluck, F., Grover, R., Miller, S., ... & Song, M. (2014). Beyond Minecraft: Facilitating computational thinking through modeling and programming in 3D. *IEEE Computer Graphics and Applications*, (3), 68-71.
- Zorn, C., Wingrave, C. A., Charbonneau, E., & LaViola Jr, J. J. (2013). Exploring Minecraft as a conduit for increasing interest in programming. In *FDG*(pp. 352-359).
- Bayliss, J. D. (2012, September). Teaching game AI through Minecraft mods. In *Games Innovation Conference (IGIC), 2012 IEEE International* (pp. 1-4). IEEE.
- Pellas, N. (2014). Exploring Interrelationships among High School Students' Engagement Factors in Introductory Programming Courses via a 3D Multi-user Serious Game Created in Open Sim. *J. UCS*, 20(12), 1608-1628.

- Jiau, H. C., Chen, J. C., & Ssu, K. F. (2009). Enhancing self-motivation in learning programming using game-based simulation and metrics. *Education, IEEE Transactions on*, 52(4), 555-562.
- dan200. 2013. ComputerCraft | Programmable computers for Minecraft. Sivustolta: <http://www.computercraft.info> [Haettu 24.4.2016].
- Cadenhead, R. (2014). *Minecraft Mods Programming*. Pearson Education.
- Gupta, A., & Gupta, A. (2015). *Minecraft Modding with Forge: A Family-Friendly Guide to Building Fun Mods in Java*. " O'Reilly Media, Inc."
- Risberg, C. (2015). more than Just a Video Game: tips for Using minecraft to personalize the Curriculum and promote Creativity, Collaboration, and problem Solving. *IAGC JOURNAL FOCUS: CREATIVITY, CRITICAL THINKING, AND CURRICULUM*, 44.
- Cateté, V., Wassell, K., & Barnes, T. (2014, March). Use and development of entertainment technologies in after school STEM program. In *Proceedings of the 45th ACM technical symposium on Computer science education* (pp. 163-168). ACM.
- Saito, D., Washizaki, H., & Fukazawa, Y. (2015, December). Work in progress: A comparison of programming way: Illustration-based programming and text-based programming. In *Teaching, Assessment, and Learning for Engineering (TALE), 2015 IEEE International Conference on* (pp. 220-223). IEEE.
- Houser, R., & DeLoach, S. (1998). Learning from games: Seven principles of effective design. *Technical Communication*, 45(3), 319-330.