

Lappeenrannan teknillinen yliopisto
School of Business and Management
Tietotekniikan koulutusohjelma

Kandidaatintyö

Oskari Liukku

**PELIOHJELMOINTIKURSSIN SUUNNITTELU
KYSELYTUTKIMUKSEN AVULLA**

Työn tarkastaja: Tutkijatohtori Jussi Kasurinen

Työn ohjaaja: Tutkijatohtori Jussi Kasurinen

TIIVISTELMÄ

Lappeenrannan teknillinen yliopisto
School of Business and Management
Tietotekniikan koulutusohjelma

Oskari Liukku

Peliohjelmointikurssin suunnittelu kyselytutkimuksen avulla

Kandidaatintyö

2017

47 sivua, 7 kuvaa, 5 taulukkoa, 2 liitettä

Työn tarkastaja: Tutkijatohtori Jussi Kasurinen

Hakusanat: peli, ohjelmointi, kurssi, unity, lukio, kysely

Keywords: game, programming, class, course, unity, high school, questionnaire

Pelit ovat tulleet osaksi kaikenikäisten suomalaisten arkipäivää. Pelialan kasvaessa Unity-pelimoottori on noussut yhdeksi alan suosituimmista työkaluista. Tässä työssä järjestettiin kysely Helsingin Suomalais-venäläisen koulun lukiolaisille. Kyselyn tavoitteena oli selvittää opiskelijoiden taitotasoa sekä toivomuksia Unity-peliohjelmointikurssille. Tulokset vastasivat kansainvälisten tutkimusten tuloksia. Lukiolaisilla ei voi kurssin alussa olettaa olevan ohjelmointitaitoja. Kurssilla tulisi opettaa ohjelmoinnin lisäksi myös muita pelin tekemisen osa-alueita. Tulosten perusteella annettiin yleisiä suosituksia lukion peliohjelmointikurssin sisällölle. Lisäksi LUMA-keskus Saimaan tuottaman Unity-peliohjelmointikurssin rakenne muokattiin suositusten mukaiseksi.

ABSTRACT

Lappeenranta University of Technology
School of Business and Management
Degree Program in Computer Science

Oskari Liukku

Designing a high school game programming course using a survey

Bachelor's Thesis

2017

47 pages, 7 figures, 5 tables, 2 appendices

Examiners: D.Sc. (Tech.) Jussi Kasurinen

Keywords: game, programming, class, course, unity, high school, survey

Games have become part of many Finns' everyday lives. As the games industry has grown, the Unity game engine has become one of the most popular tools in the industry. For this thesis a survey was arranged for the students of the Finnish-Russian School of Helsinki. The goals of the survey were to find out about the students' skill levels as well as what they would like to see on an Unity game programming course. The results corresponded with results from international studies. The students should be assumed to have no initial programming skills at the beginning of the course. In addition to programming, the course should teach other aspects of game making too. The results of the survey were used to generate general recommendations for organizing a high school programming course. Additionally, a course structure for a Unity game programming course, used by LUMA Centre Saimaa, was modified to comply with the recommendations.

ALKUSANAT

Työ tehtiin Lappeenrannan teknilliselle yliopistolle. Työn aihe saatiin tutkijaopettaja Uolevi Nikulalta. Haluan esittää kiitokset Jussi Kasuriselle työn valvomisesta, ja sekä Kasuriselle että Ari Haposelle erinomaisesta rakentavasta palautteesta työn aikana. Haluan kiittää Helsingin Suomalais-venäläisen koulua haastatteluluvasta. Kiitän myös perhettäni sekä Max Samarinia ja Erika Anttilaa avusta oikolukemisessa.

SISÄLLYS

1. JOHDANTO	7
1.1 TYÖN TAUSTA	7
1.2 TAVOITTEET JA RAJAUKSET	8
1.3 TYÖN RAKENNE	9
2. KIRJALLISUUSKATSAUS	10
2.1 MITÄ PELIOHJELMOIJAN TULISI OSATA	10
2.2 OHJELMOINTIKURSSIN RAKENNE	11
2.3 OHJELMOINNIN OPETUKSEN STANDARDIT	12
2.4 PELINKEHITYS OHJELMOINNIN OPETUKSEN TYÖKALUNA	13
2.5 SISÄÄNRAKENNETUT INTERAKTIIVISET OPPITUNNIT	14
2.6 ILMAISET VERKKOKURSSIT	16
3. UNITY-PELIMOOTTORI	19
4. KYSELY LUKIOLAISILLE	21
4.1 DATAA VASTAAJISTA	21
4.2 VASTAAJIEN MIELIPITEITÄ KURSSIN SISÄLLÖSTÄ	22
4.3. JOHTOPÄÄTÖKSIÄ KYSELYN TULOKSISTA	24
5. KURSSIN MUOVAAMINEN KYSELYN PERUSTEELLA	27
5.1 YLEISIÄ SUOSITUKSIA	27
5.2 KURSSISUUNNITELMA LUMA:N KURSSIA POHJANA KÄYTTÄEN	28
6. YHTEENVETO	33
7. LÄHTEET	34

LIITTEET

1. Kysely lukiolaisille
2. Yhteenveto kyselyn vastauksista

SYMBOLI- JA LYHENNELUETTELO

ACM	Association of Computing Machinery
CSTA	Computer Science Teachers Association
ESA	Entertainment Software Association
LUT	Lappeenrannan Teknillinen Yliopisto
SVK	Helsingin Suomalais-venäläinen koulu

1. JOHDANTO

Peliteollisuuden sanotaan jo vuosia olleen räjähdysmäisessä kasvussa Suomessa. Pelit ovat tulleet osaksi arkipäivää suurelle osalle nuorista, ja monelle aikuisellekin — vuonna 2015 suomalaisista 87.8 % pelasi aktiivisesti [1]. Alan kasvaessa on kiinnostus sen opiskelua ja harrastamista kohtaan lisääntynyt. Unity-pelimoottori on noussut alan kärkeen pelien kehityksen työkaluna niin aloittelijoiden kuin ammattilaistenkin keskuudessa. VisionMobilen vuoden 2004 kyselyn mukaan 29 % pelinkehittäjistä käytti sitä päätyökalunaan, kun kilpailijapelimoottoreita kuten Cocos2d:ta ja Unreal Engineä käytti vain 8 % ja 6 %. [2] Vuonna 2016 34 % ilmaisista mobiilipeleistä tehtiin Unityllä [3]. Tässä työssä tarkastellaan Unityn käyttämistä peliohjelmoinnin opetuksessa.

1.1 Työn tausta

Opinnäytetyö tehdään Lappeenrannan teknilliselle yliopistolle (LUT). Työn pohjana käytetään LUT:n yhdessä LUMA-keskus Saimaan (luonnontieteiden, matematiikan ja teknologian opetusta toteuttava keskus) kanssa tuottamaa peliohjelmoinnin lukio-opetuksen kurssia [4], jota muokataan kyselyn tulosten perusteella.

Suomen opetushallitus hyväksyi ohjelmoinnin perusopetuksen opetussuunnitelmaan joulukuussa 2014. Opetus alkaa ensimmäiseltä luokalta ja jatkuu läpi peruskoulun. Peruskoulun käytyään oppilas osaa soveltaa algoritmista ajattelua ja ohjelmointia ongelmien ratkaisemiseen, on tutustunut hyviin ohjelmointikäytäntöihin, sekä osaa ohjelmoida yksinkertaisia ohjelmia. [5] Lukiossa ohjelmoinnin opiskeleminen on valinnaista, eivätkä kaikki lukiot järjestä ollenkaan ohjelmointikursseja. Sen sijaan kiinnostuneille lukiolaisille suositellaan yliopistojen järjestämiä kursseja ja verkkokursseja.

1.2 Tavoitteet ja rajaukset

Tämän työn tavoitteena on haastatella lukiolaisia ja määrittää heidän teknistä lähtötasoaan sekä odotuksiaan ja toiveitaan peliohjelmointikurssille. Niiden perusteella esitetään suosituksia lukion peliohjelmointikurssin sisällölle sekä muokataan LUMA-keskus Saimaan kurssirakenne tulosten mukaiseksi. Luentojen ja tehtävien rinnalla kurssilla tehdään yksinkertainen esimerkkipeli, jonka tavoitteena on saada opiskelijat huomaamaan, miten kurssilla opittuja taitoja voidaan soveltaa käytäntöön. Kurssin loppupää on omistettu pelin ohjatulle viimeistelylle, jolloin jokainen opiskelija saa muokata pelistä haluamansalaisen.

Ohjelmointikurssin tavoitteena on perehdyttää opiskelija Unity-pelimoottoriin riittävän hyvin, jotta hän voi halutessaan jatkaa pelien kehittämistä itsenäisesti kurssin jälkeen. Siksi yksi tärkeä kurssilla painotettava asia on itsenäinen tiedonhaku ja sen menet. Kurssin ohjelmointikielenä käytetään C#:a. Unity-pelimoottori tukee myös Javascriptin kaltaista UnityScript-ohjelmointikieltä, mutta Unity on lopettamassa sen tukemisen [6]. Opiskelija oppii kurssilla siis pelinkehityksen lisäksi myös C#:n perusteet ja yleisesti ohjelmoinnissa käytettävää logiikkaa ja ajattelutapoja.

Tutkimuskysymys on kaksiosainen:

- *Mitä lukiolaiset toivovat peliohjelmointikurssilta?*
- *Miten toivomuksia voidaan hyödyntää ohjelmointikurssin suunnittelussa?*

Työssä käytetään Unityn versiota 2017.2.0f3. Unity julkaisee vuosittain uuden version, johon se vuoden mittaan päivittää uusia pienempiä ominaisuuksia ja korjauksia. Vuoden 2017 versiot on merkitty tunnisteella 2017.X, jossa X on pienemmän päivityksen numero. Työhön valittiin versio 2017.2 sen mukana tulleen uudenlaisen interaktiivisen oppituntiominaisuuden vuoksi.

1.3 Työn rakenne

Työ alkaa kirjallisuuskatsauksella, jossa perehdytään aiheesta jo tehtyihin tutkimuksiin ja materiaaliin, sekä muutamaaan ilmaiseen internetistä löytyvään ohjelmointikurssiin. Tämän jälkeen kuvaillaan lyhyesti Unity-pelimoottorin toimintaa ja osia. Opiskelijoiden haastattelun suunnittelusta, haastatteluprosessista ja haastatteluiden tuloksista kerrotaan. Tulosten perusteella esitetään suosituksia lukion peliohjelmointikurssille, sekä muokataan LUMA-keskus Saimaan kurssirakenne tulosten mukaisesti. Lopuksi on yhteenveto työn tuloksista, pohdintaa ja jatkotutkimusehdotuksia.

2. KIRJALLISUUSKATSAUS

Kirjallisuuskatsausta varten haettiin peliohjelmoinnin ilmaisia aloituskursseja ja -oppitunteja Googlen avulla. Koska työssä kyse on Unity-kurssista, keskityttiin pääasiassa Unityä opettavaan aineistoon, mutta myös muut lähteet otettiin vertailun vuoksi huomioon. Tämän lisäksi tässä luvussa tarkastellaan työn kannalta tärkeitä, jo tehtyjä tutkimuksia ja aineistoa, joita haettiin Google Scholarin ja Association of Computing Machinery:n (ACM) sivujen hakutoiminnon kautta. Erityisesti haettiin tietotekniikan tai peliohjelmoinnin kursseille tehtyjä ohjeita tai sääntöjä. Käytettyjä hakusanoja olivat muun muassa *games*, *programming*, *teaching*, *unity*, *unity3d*, *course*, *class*, *curriculum* ja *guideline*.

2.1 Mitä peliohjelmoijan tulisi osata

Professorit Manjula, Chakradhar Raju ja Sai Chand selvittivät artikkelissaan *Software Engineering Challenges in Game Development* [7] pelin tekemiseen vaadittavia ohjelmointitaitoja. He huomauttavat, että nykyaikaisissa peleissä saatetaan vaatia monen tieteenalan, esimerkiksi fyysikan, 3D-grafiikan, tekoälyjen ja verkkoarkkitehtuurin, osaamista. Suurissa peliyrityksissä ohjelmoijat jaetaan tiimeihin heidän erikoistumisensa mukaan, jolloin tietojen jakaminen helpottuu.

Jonathan Blowin kertoo artikkelissaan *Game Development: Harder Than You Think* [8] nykyajan pelinkehityksen olevan paljon monimutkaisempaa ja vaikeampaa kuin 1990-luvulla. Esimerkiksi simulaatiot ovat nykyään hyvin tarkkoja verrattuna pelinkehityksen alkuaikoihin. Projektikoko on muun muassa palvelimien tarpeellisuuden ja alustojen määrän takia moninkertaistunut.

Opetushallitus haastatteli vuoden 2016 raportissaan [9] 19:sta pelialan yritystä — yhteensä 492:ta alan ammatilasta. Haastatteluissa selvitettiin muun muassa millaisia taitoja alalla tarvitaan. 10 suurinta osaamistarvetta on listattu taulukossa 1.

Taulukko 1. Vuoden 2016 pelialan suurimmat osaamistarpeet [9].

Osaamistarve	Työntekijöitä	Yhtiöitä
Unity-pelimoottori	106	14
C++-peliohjelmointi	100	9
2D-grafiikan tuottaminen	68	5
Adobe Flash -kehitysympäristö	67	6
Kenttäsuunnittelu	48	8
Windows Phone -käyttöliittymä	48	2
C#-peliohjelmointi	47	6
Palvelin- ja palvelinjärjestelmäohjelmointi	45	2
3D-mallien tekstuurigrafiikka	43	8
Apple iOS -käyttöjärjestelmä	42	3

2.2 Ohjelmointikurssin rakenne

ACM julkaisi vuonna 2013 oppaan [10] tietotekniikan opetussuunnitelmille. ACM huomautti oppaassaan, että opetuksessa on tärkeää huomioida alan monihaaraisuus. Pelien lisäksi opiskelija voi erikoistua kymmenille muille suuntauksille, jotka kaikki kuuluvat tietotekniikan nimikkeelle. Tämän takia on tärkeää valmistella paljon valinnaisia tehtäviä, jotta opiskelija voi seurata omia kiinnostuksiaan perusasiat opittuaan.

ACM kertoo oppaassaan myös, että suuntauksesta riippumatta opiskelijat oppivat soveltamaan oppimaansa parhaiten tekemällä käytännön ohjelmointiprojekteja. Projekteilla opiskelijat voidaan

jakaa tasaisesti ryhmiin ja ryhmille voidaan antaa sopivan haastava tehtävä. Näin opiskelijat oppivat kommunikoimaan ja toimimaan tehokkaasti ryhmässä, mikä on ACM:n mukaan tietotekniikan alalla elintärkeää.

Opas sisältää opintosuunnitelman ja kurssisuositukset 18:sta aihealueelle. Oppaaseen on myös haastateltu useita amerikkalaisten yliopistojen tietotekniikan haarojen edustajia.

2.3 Ohjelmoinnin opetuksen standardit

Yhdysvaltain tietotekniikan opettajien liitto eli Computer Science Teachers Association (CSTA) on yhteistyössä ACM:n kanssa julkaissut listan tietotekniikan opetuksen standardeja [11]. Se listaa standardien päätavoitteeksi kattavan tietotekniikan opetuksen tarjoamisen kaikille opiskelijoille. Standardien avulla opettajien on mahdollista asettaa yksityiskohtaisia ja mitattavia tavoitteita opetukseen.

Taulukko 2. CSTA:n standardeja lukion tietotekniikan opetukselle (mukaillen käännetty) [11].

Standardi	Käsite	Alakäsite	Aihealue
Ohjelmistojen kategorisointi rooleihin	Tietojärjestelmät	Laitteisto ja ohjelmisto	Kommunikaatio
Tapojen, joilla tietojärjestelmät toteuttavat loogiikkaa sekä tiedon syöttöä ja tulostusta, kuvaileminen	Tietojärjestelmät	Ongelmanratkaisu	Kommunikaatio
Verkon toimivuuteen vaikuttavien seikkojen (esim. kaistanleveys, taakka, viive) kuvaileminen	Tietoverkot ja Internet	Verkkokommunikaatio ja -organisaatio	Kommunikaatio
Tapojen, joilla	Tietoverkot ja	Tietoturva	Kommunikaatio

ohjelmistokehittäjät suojaavat laitteita ja dataa luvattomalta käytöltä, vertailu	Internet		
Data-analyysityökalujen ja -tekniikkojen käyttäminen kaavojen löytämiseen monimutkaisten systeemien datasta	Data-analyysi	Datan visualisointi ja transformointi	Abstrahointi

Standardeja on yhteensä 120, esikouluiästä lukioon. Niistä 28 on tarkoitettu lukiotasolle (16-18 vuotiaille). Taulukkoon 2. on käännetty viisi ensimmäistä lukiostandardia. CSTA:n listassa jokaiselle standardille on otsikon lisäksi tarkempi kuvaus, ja osalle aihealueista myös esimerkkitehtäviä.

2.4 Pelinkehitys ohjelmoinnin opetuksen työkaluna

Mohammed Al-Bow et al. julkaisivat vuonna 2009 tutkimuksen [12], jossa he kuvailivat *P4Games* (Pixels, Programming, Play & Pedagogy) -projektiaan ja sen tuloksia. Projektin tarkoituksena oli tutustuttaa nuoria ohjelmointiin pelien avulla. Projektiin sisältyi kahden viikon ohjelmointikesäleiri 14-15 -vuotiaille koululaisille, neljän viikon opettajakoulutus peliohjelmoinnin opettamista varten, sekä resursseja opettajille ja kouluille opetussuunnitelman implementoimisen avuksi.

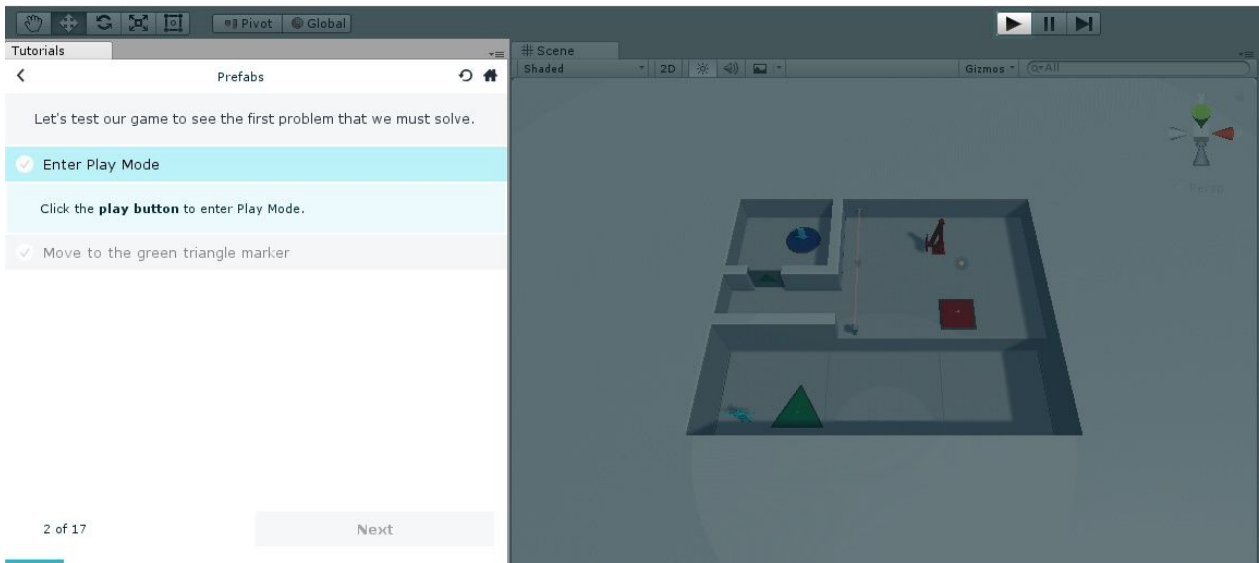
Nuorten kesäleirin kymmenestä päivästä kuusi ensimmäistä olivat koulutusta ja loput neljä vapaata oman pelin kehitystä. Koulutuspäivät oli jaettu neljään oppituntiin: ohjelmointi, taide, pelisuunnittelu ja leikki tai pelailu. Taidetunnit etenivät kolmiulotteisista rautalankanukeista hiljalleen digitaaliseen taiteeseen. Aiheita olivat muun muassa erilaiset perspektiivit, sommitelmat ja valo ja varjo. Pelisuunnittelussa nuoret suunnittelivat pelejä sekä opettelivat antamaan hyvää rakentavaa kritiikkiä muiden peleistä ja parantelemaan omiaan saadun palautteen

perusteella. Ohjelmoinnissa käytettiin kielenä Javaa ja työkaluna Greenfoot-ohjelmointiympäristöä. Opiskelijoiden annettiin ensiksi tutustua Greenfootiin valmiin pelin turvin, jonka jälkeen luotiin yksinkertainen peli askel kerrallaan. Ohjelmoinnin käsitteet opetettiin pelinkehityksen kannalta loogisessa järjestyksessä: kuvien lisääminen ja liikuttelu, kuvien kimpoaminen pelialueen reunoista (ehtolauseet ja datan käsittely), kirjainten muodostaminen kuvista (koordinaatit ja for-silmukat), näppäimistön käyttäminen kuvien liikutteluun, yksinkertainen tekoäly, ja pelin kello (objektit ja niiden jäsenet).

Ennen ja jälkeen vuoden 2008 kurssien tehtiin kyselyitä, joilla testattiin kurssille osallistuvien lähtötasoja, oppimista sekä itsevarmuuden kehittymistä. Testattavia alueita olivat muun muassa for-silmukat, muuttujat ja ehtolausekkeet. Ohjelmointikyselyssä vastausvaihtoehdot olivat kyllä, ei ja en tiedä. Leirin lopuksi kyllä-vastauksien määrä oli kasvanut huomattavasti, ja erityisesti en tiedä -vastaukset olivat vähentyneet. Al-Bowin et al. (2009) mukaan aloittelijoiden itsevarmuuden ja kiinnostuksen kasvattaminen on jopa tärkeämpää kuin täysin oikeiden vastauksien hakeminen. [12]

2.5 Sisäänrakennetut interaktiiviset oppitunnit

Unity esitteli versiossa 2017.2 pelimoottoriin sisälle rakennetut oppitunnit [13]. Oppitunnit tutustuttavat käyttäjän Unityn käyttöliittymään ja perusominaisuuksiin. Samalla pieneen esimerkkipeliin lisätään lisäominaisuuksia, ja oppija pääsee kokeilemaan tekemiään muutoksia saman tien nähdäkseen niiden vaikutuksen. Oppitunteja varten Unityyn on lisätty myös joitakin ääniefektejä.



Kuva 1. Unityn oppituntinäkömä.

Kuvassa 1. on Unityn oppituntinäkömä. Näkömän vasempaan reunaan päivittyvät ohjeet sitä mukaa, kun käyttäjä suorittaa oppituntia. Kuvan oikealla puolella on kohtausnäkömä (engl. scene view), jossa pelin objekteja voi siirrellä. Oppitunti korostaa relevantit osat ruudusta keskittymisen helpottamiseksi. Koska kuvassa on työvaiheena Play-moodiin siirtyminen, vain ohjeet ja Play-painike on valaistu.

Ensimmäisessä erässä oppitunteja julkaistiin neljä:

1. *Play & Edit Mode* esittelee editorin eri tiloja
2. *Game Objects and Components* selittää, mitä peliobjektit (engl. Game Object) ovat, sekä esittelee komponentit (engl. component)
3. *Tweaking Components* esittelee miten komponenttien arvoja voi muokata editorissa
4. *Prefab power* esittelee prefabit (valmiiksi kasatut pelielementit) ja niiden käytön

Nämä neljä oppituntia esittelevät vain hyvin alkeellisia asioita Unityn käytöstä. Unity on luvannut tuottaa käyttäjien palautteen perusteella lisää oppitunteja, sekä julkaista työkalut oppituntien luomiseen kaikkien saataville. [13]

2.6 Ilmaiset verkkokurssit

Internetissä on satoja laadukkaita, ilmaisia peliohjelmointikursseja. Alla on taulukoitu kolmen suuren Unity-verkko-oppituntien tuottajan sekä yhden yleisesti peliohjelmointia opettavan sivuston ominaisuuksia. Tarkasteltavia kursseja haettiin Googlen avulla, hakusanoilla *unity, coding, programming, tutorial, free*. Valintakriteereinä olivat ammattimaisuus ja sisällön suunnittelun laatu, jotta kurssin sisältö olisi helppo kääntää myös suomen kielelle. Tämän takia joukossa ei ole muita videomuotoisia oppitunteja kuin Unityn oma. Taulukoidut kävijämäärät ovat SimilarWeb -palvelun [14] avulla laskettuja arvioita sivuston käyttäjämäärälle kuukaudessa. Tämä luku on kuitenkin erityisesti Unityn kohdalla harhaanjohtava, sillä siihen on laskettu koko osoitteen kävijämäärä, ei ainoastaan oppituntisivujen.

Taulukko 3. Erilaisten aloittelijakurssien ominaisuuksia.

	Unity	Catlike Coding	Noobtuts	Game Code School
Maksullisuus	Ilmainen	Ilmainen	Osittain	Osittain
Kävijämäärä	24 000 000	400 000	Ei saatavilla	170 000
Pelimoottori	Unity	Unity	Unity	-
Laajuus	9 peliprojektia, 500+ muuta oppituntia	65+ tekstioppituntia	19 peliprojektia, joista 8 ilmaisia	100+ tekstioppituntia
Ohjelmointikieli	C#	C#	C#	Java
Esitysmuoto	Video	Teksti	Teksti	Teksti
Ensimmäinen projekti	Pallopeli	Liikkuva kello	Pong-peli	Pong-peli

Unity on tuottanut satoja oppitunteja omalle pelimoottorilleen. [15] Ne ovat kaikki ilmaiseksi saatavilla yhtiön verkkosivujen Learn-osastolla. Oppitunneista suuri osa on videomuodossa, mutta osa haastavampia aiheita käsittelevistä artikkeleista on toteutettu vain tekstinä ja kuvina. Unityn suosittelee aloittelijalle *Roll-a-ball tutorial* -projektia [16], joka koostuu yhdeksästä video-oppitunnista. Projektin tuloksena syntyy pieni esimerkkipeli, jossa pelaaja voi liikuttaa palloa maailmassa käyttäen Unityn fysiikkamoottoria ja keräillä pisteitä kerryttäviä esineitä.

Catlike Coding on Jasper Flickin perustama yritys, joka tuottaa muiden projektien rinnalla ilmaisia Unity-oppitunteja samannimisille verkkosivuille. [17] Catlike Coding-sivusto pitää sisällään kymmeniä kirjallisia Unity-oppitunteja [18], jotka käsittelevät aiheita alkeista asiantuntijatasolle. Vaikka oppitunnit ovat kirjallisia, ne on onnistuttu pitämään samalla sekä yksityiskohtaisina että lyhyehköinä sisällyttämällä tekstin sekaan laatikoita, joita klikkaamalla lukija saa pätkän lisätietoa aiheesta. Flick käyttää myös paljon havainnollistavia kuvia ja animaatioita tekstin seassa. Ensimmäinen oppitunti sivustolla on *Game Objects and Scripts* [19], joka on yksi viidestä oppitunnista *Basics*-otsikon alla. Oppitunti on projekti, jonka aikana rakennetaan Unityssä yksinkertainen kello ja oikean ajan mukaan liikkuvat viisarit. Samalla opitaan Unityn ja C#:n alkeita.

Noobtuts on vuonna 2012 perustettu sivusto, joka tekee Unityn lisäksi aloittelijaoppitunteja myös muun muassa C++ ja Python -ohjelmointikielille. Se kertoo tunnuslauseensa olevan “Puhu minulle kuin olisin viisivuotias” [20] eli se pyrkii tekemään aloittelijoille helposti lähestyttäviä oppitunteja. Sivuston Unity-oppitunnit [21] keskittyvät klassikkopelien kuten Pong, Pac-Man ja Tetris, jäljittelyyn. Osa vaikeammista projekteista on maksullisia. Noobtutsin Pong-oppitunti on oppitunneista helpoin. Yksityiskohtaisesti kuvatun ja kuvilla havainnollistetun oppitunnin aikana lukijalle opetetaan Unityn ja C#:n perusteiden lisäksi myös hieman pelisuunnittelua.

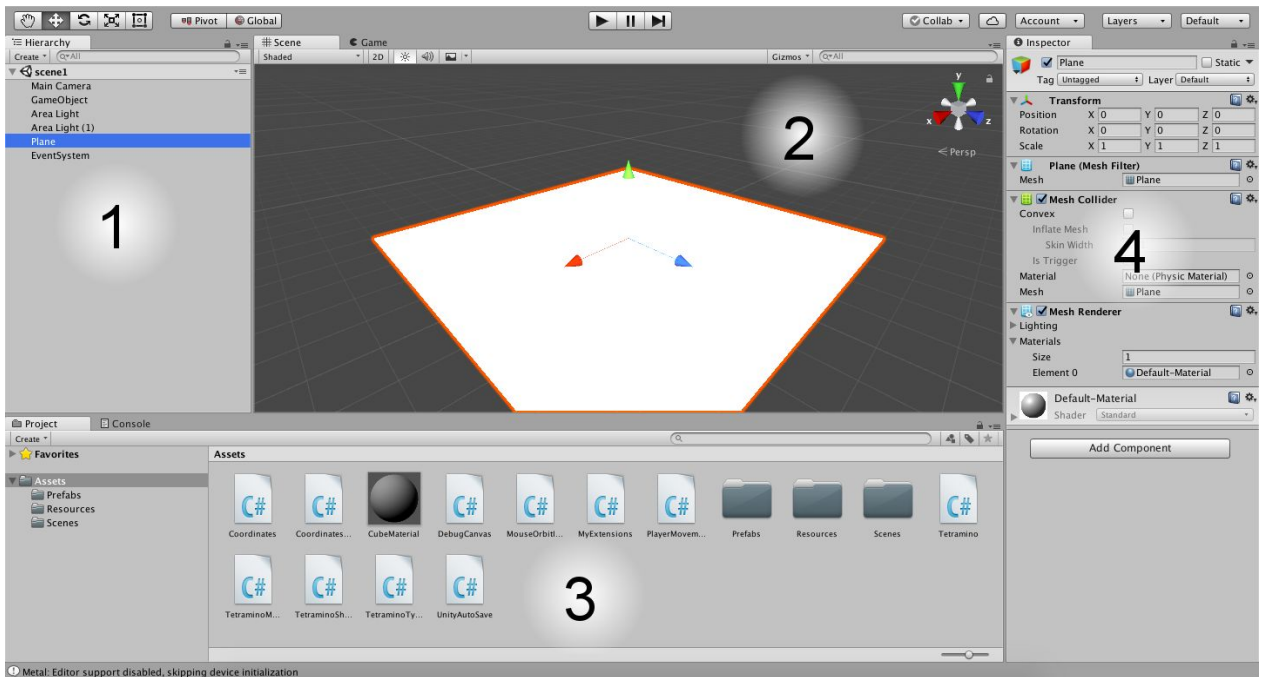
Game Code School [22] on sivusto, joka tarjoaa ilmaisia oppitunteja useilla eri ohjelmointikielillä, pääosin Javalla. Sivustolla on koodioppituntien lisäksi myös *Game Essentials*

-osio, joka keskittyy opettamaan peliohjelmoinnin tärkeimpiä teoria-aihealueita. Java-oppitunteja on aloittelijasta ammattilaistasolle. Alkeiden opettamisen jälkeen siirrytään pelien kehittämiseen Android-puhelimille. [23,24] Osa kurssin moduuleista on maksullisia (esimerkiksi ensimmäisen oikean pelin teko-ohjeet löytyvät e-kirjasta), mutta ilmaisia aloittelijoille suunniteltuja Java-peliprojekteja löytyy sivustolta useita.

3. UNITY-PELIMOOTTORI

Unity on pelimoottori, joka on muun muassa ilmaisuutensa ja laajan alustatukensa ansiosta noussut yhdeksi maailman suosituimmista. Unity julkaistiin vuonna 2005 Apple Mac-tietokoneille [25] ja heti seuraavana vuonna se pääsi toiselle sijalle Applen Design Awards -palkintogaalassa. [26] Vuonna 2015 Unity yhdisti pelimoottorin Free- ja Pro-lisenssit siten, että kaikki moottorin ominaisuudet tulivat ilmaiseen käyttöön. [27] Unityn kilpailija, Epic Games -yrityksen oma pelimoottori Unreal Engine, oli edellisenä päivänä ilmoittanut muuttuvansa ilmaiseksi. [28] Vaikka itsenäisesti tuotettujen pelien eli *indiepelien* tekeminen oli jo ennen tätä ollut kasvussa, Unityn ja Unrealin maksuttomaksi muuttumisen sanotaan tuoneen pelinkehityksen todella kaikkien saataville.

Unityllä tehdyt pelit koostuvat kohtauksista (engl. scene) — esimerkiksi päävalikko, erilaiset tasot, lopputekstit — jotka koostuvat edelleen peliobjekteista (engl. game object) ja muusta datasta. Peliobjektit sisältävät eri määriä komponentteja (engl. component) jotka ovat instansseja eli ilmentymiä (engl. instance) komentosarjoista eli skripteistä (engl. script). Kaikilla kohtauksen objekteilla on *Transform*-komponentti, joka sisältää dataa muun muassa objektin sijainnista, koosta ja nivelpisteestä (engl. pivot). Unity sisältää satoja valmiita komponentteja ja skriptejä joita voi kiinnittää objekteihin, mutta niitä voi myös koodata itse. Näin voidaan toteuttaa pelin logiikka.



Kuva 3. Unityn käyttöliittymä

Kuvassa 3 on numeroitu Unityn käyttöliittymän pääalueet. Kohta 1 on kohtauksen hierarkia (engl. hierarchy), jossa näkyy listattuna kaikki kohtauksessa olevat objektit. Kohta 2 on kohtauksen esikatselunäkymä. Valkoinen neliö on valittuna, joten se näkyy sekä kohtauksessa että hierarkiassa korostettuna. Valittua objektia voi liikutella joko esikatselunäkymän kautta hinaamalla tai kuvassa näkyvien nuolien avulla. Kohta 3 on projekti-ikkuna. Sen vasemmalla puolella on projektin pääkansion kaikki alakansiot ja -tiedostot, ja keskellä näkyy valitun kansion sisältö. Kohta 4 on tarkastelunäkymä (engl. inspector), joka näyttää valitun objektin tiedot. Kuvassa valittuun objektiin on kiinnitetty komponentit *Transform*, *Mesh Filter*, *Mesh Collider* ja *Mesh Renderer*.

4. KYSELY LUKIOLAISILLE

Kysely järjestettiin 19.12.2017 Helsingin Suomalais-venäläisen koulun (SVK) opiskelijoille. Kyselyä mainostettiin koulun kirjastossa kyltillä, ja kaikki kiinnostuneet lukiolaiset ja yhdeksäsluokkalaiset saivat linkin kyselyyn. Lisäksi osallistujia houkuteltiin makeispalkinnoilla. Kysely oli kirjallinen, mutta vastaamisen aikana kysyttiin myös suullisia lisäkysymyksiä ja mielipiteitä, sekä tarkkailtiin vastaajien reaktioita kysymyksiin. Kyselyyn osallistui 24 opiskelijaa sekä yksi opettaja. Opettaja toi koko ryhmänsä osallistumaan projektiin. Kyselylomake löytyy kokonaisuudessaan liitteestä 1, ja yhteenveto saaduista vastauksista liitteestä 2.

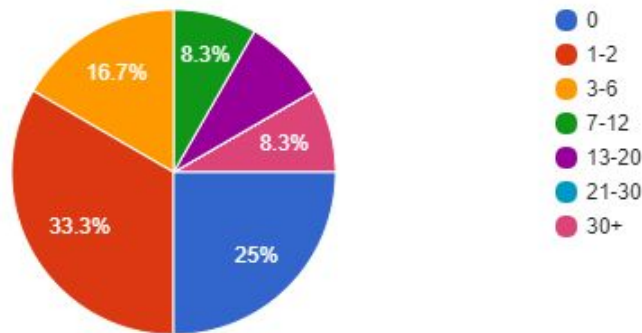
4.1 Dataa vastaajista

Kyselyyn vastasi jokaisen vuosikurssin edustajia. Suurimman osan vastaajajoukosta muodostivat 15- ja 18-vuotiaat eli 9.-luokkalaiset ja abiturientit. Joukosta vain kaksi opiskelijaa (8.3 %) kertoi osaavansa jo ohjelmoida. Heistäkin toinen vastasi jatkokysymykseen hallituista ohjelmointikielistä “suomi/venäjä”, mikä kielii väärinymmärryksestä. Kahdeksan vastaajaa kertoi kuullensa joskus Unity-pelimoottorista, ja viisi kertoi kokeilleensa sitä. Muutama kertoi myös kuulleensa Unreal Enginestä ja kysyi, käytetäänkö kurssilla sitäkin.

Peliharrastajia vastaajien joukosta löytyi runsaasti — vain 25 % kertoi pelaavansa 0 tuntia viikossa. Kolmasosa sanoi pelaavansa 1-2 tuntia viikossa, ja 8.3 % vastasi pelaavansa yli 30 tuntia viikossa. Koko jakauma on näkyvässä kuvassa 4. Vain muutama vastaajista myönsi pelaavansa kun heiltä sitä kysyttiin kasvotusten, ja sen sijaan moni kääntyi osoittamaan luokkalaistaan ja syyttämään tätä innokkaammaksi pelaajaksi.

Montako tuntia viikossa keskimäärin pelaat/katsot/harrastat videopelejä?

24 responses



Kuva 4. Lukiolaisten viikottainen pelaaminen (tuntia viikossa)

Älypuhelin oli ehdottomasti suosituin pelialusta — 62.5 % vastaajista listasi sen yhdeksi pelilaitteistaan. Seuraavaksi tuli tietokone 42.6 %:lla ja kolmanneksi konsolit 33.3 %:lla.

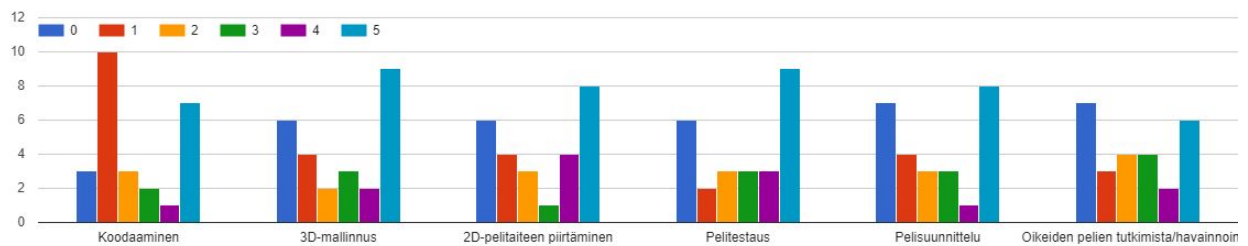
Kyselyn toinen osio sisälsi tarkentavia kysymyksiä vastaajien pelinkehitystaidoista. Taitojen aihealueita olivat 3D-mallien luonti, piirtäminen, pelisuunnittelu ja pelitestausta. 25 % kertoi osaavansa luoda 3D-malleja tietokoneella jollain tasolla. 70.8 % sanoi osaavansa piirtää tietokoneella jollain tasolla. Moni kysyi kysymykseen törmätessään, että onko Microsoft Paint riittävän taidokasta piirtämistä. Kysyjille vakuutettiin, että Microsoft Paint riittää hyvin pelitaitteen tekemiseen. 12.5 % on joskus suunnitellut omaa peliä, ja 20.8 % on joskus etsinyt pelaamistaan peleistä vikoja tarkoituksenomaisesti.

4.2 Vastaajien mielipiteitä kurssin sisällöstä

Kyselyn kolmannessa vaiheessa selvitettiin kysyjien mielipiteitä kurssin mahdollisista aiheista. Kaikissa kysymyksissä, joissa pyydettiin arvioimaan kiinnostusta, vastausvaihtoehdot olivat 0-5. 5 merkitsi äärimmäistä kiinnostusta, ja 0 sitä, että kiinnostusta ei ollut ollenkaan. Vaihtoehtoja annettiin kuusi täysin neutraalien vastausten vähentämiseksi.

Kuvassa 5 on esitetty vastaajien mielipiteet kuudesta mahdollisesta kurssin aiheesta: koodaaminen, 3D-mallinnus, 2D-piirtäminen, pelitestausta, pelisuunnittelu, oikeiden pelien havainnointi. Vastaukset ovat melko polaarisia — noin puolet ovat todella kiinnostuneita ja puolia ei aihe kiinnosta ollenkaan. Kun henkilöt joita mikään aihealueista ei kiinnosta (yli viisi 0-1 -arvosanaa) karsitaan pois, koodaamisen keskiarvosanaksi tulee 3.1, 3D-mallinnuksen sekä 2D-taiteen 3.8, pelitestaamisen 3.9 ja pelisuunnittelun sekä pelien havainnoinnin 3.38. Ei-kiinnostuneet vastaajat karsittiin keskiarvolaskelmista pois, sillä opiskelija, joka ei ole kiinnostunut mistään kurssin aiheesta, ei kuulu kurssin kohdeyleisöön.

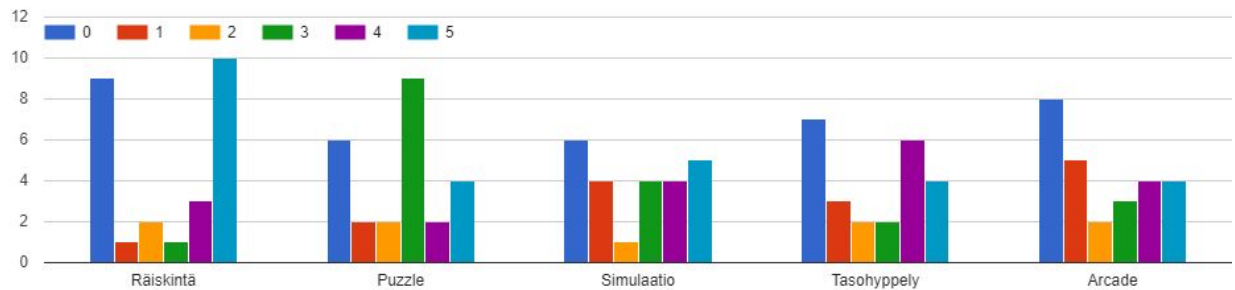
Kuinka kiinnostunut olet seuraavista mahdollisista kurssin aiheista (0 = en yhtään, 5 = todella kiinnostunut)?



Kuva 5. Lukiolaisten kiinnostus pelinkehityksen eri aihealueisiin

Lopuksi kysyttiin, mille laitteelle ja millaista peliä lukiolaiset haluaisivat kurssilla mieluiten tehdä. Laitteista selvästi suosituin oli tietokone 70.8 %:lla äänistä. Toiseksi eniten ääniä sai älypuhelin 37.5 %:lla ja kolmanneksi selainpeli 29.2 %:lla. Pelistä halutaan 3D-peli 87.5 %:n osuudella äänistä.

Minkä tyylistä peliä haluaisit tehdä kurssilla (0 = ei kiinnostu, 5 = todella kiinnostunut)?



Kuva 6. Lukiolaisten arvosanoja kurssin pelin tyyliä vaihtoehtojen

Kuvassa 6 on pelin tyyliä -kysymyksen vastauksia. Eniten 5-arvosanoja sai räiskintäpeli (josta annettiin esimerkkeinä Counter Strike ja Overwatch -pelit), mutta se sai myös eniten 0-arvosanoja. Se on suosituin myös keskiarvojen perusteella; sen keskiarvo on 3.5, kun toiseksi tulevan simulaatio-vaihtoehdon keskiarvo on 2.9. Vähiten matalia arvosanoja (0-1) saanut tyyli oli kuitenkin pulmapeli (esimerkkeinä Candy Crush ja Tetris).

4.3. Johtopäätöksiä kyselyn tuloksista

Ohjelmointi on edelleen harvinainen taito lukiolaisten keskuudessa ainakin lukioissa, joissa sitä ei ole mahdollista opiskella. Kiinnostus pelinkehitykseen sekä oma-aloitteisuus ovat kuitenkin suhteellisen suuria Unity-vastausten perusteella.

Moni vastaaja epäroi kysyttäessä heidän pelaamisensa määrää. Kyselyyn he kuitenkin valitsivat rehellisen vastauksen, mikä kertoo siitä, että pelaaminen saattaa vieläkin olla joissain piireissä noloa tai siihen liittyy muita negatiivisia mielikuvia. Kurssia tulisi siis markkinoida opiskelijoille mahdollisuutena työskennellä itsenäisesti koodin ja pelien parissa.

Vahvin kyselyssä esille tullut mielipide oli, että kurssilla tehtävän pelin tulisi olla kolmiulotteinen kaksiulotteisen sijaan. Pelin tyyllilajin tulisi olla räiskintä tai pulmapeli, aikarajoitteet huomioon ottaen.

Suuri osa vastaajista kertoi olevansa kiinnostunut monesta pelintekemisen aihealueesta, mutta muutama vastasi kaikkiin arvosanoilla 0-1. Nämä jätettiin huomiotta, sillä tällaiset henkilöt tuskin tulisivat kurssille muutenkaan. Vastaajat ovat joko melko kiinnostuneita kaikesta pelinkehitykseen liittyvästä, tai eivät yhdestäkään sen osa-alueesta. Mikään aihe ei nouse selkeäksi yleisen mielenkiinnon kohteeksi. Vastaajien joukossa on havaittavissa erikoistumista. Yksi vastaajista antoi ainoastaan pelitestaamiselle korkean arvosanan, ja muutama arvioi olevansa kiinnostunut vain 2D- ja 3D-taiteesta eikä ollenkaan koodaamisesta. Onkin pohdittava, kannattaako peliohjelmointikurssia suunniteltaessa pitää myös muut pelintekemisen aihepiirit mielessä. Tämä riippuu täysin siitä, onko muita peliaiheisia kursseja tarjolla. Jos, kuten esimerkiksi SVK:n tapauksessa, uusi peliohjelmointikurssi olisi ainoa tarjolla oleva pelialan kurssi, tulisi sen tarjota alkeita myös ohjelmoinnin ulkopuolelta. Jos kursseja on tarjolla enemmän, esimerkiksi 3D-mallinnuksen ja pelitaiteen alueilta, peliohjelmointikurssi voi keskittyä tarkemmin vain ohjelmointiin. Tärkeintä on, että opiskelijalle selviää pelialana lukuisat erikoistumismahdollisuudet ja se, että kukaan ei ole ammattitasolla kaikessa, vaan on valittava mihin perehtyy [10, 12].

Tyylilajille ei selvinnyt selkeää suosikkia. On pohdittava, tehdäänkö kurssilla projektia johon mahdollisimman moni suostuu vai jota mahdollisimman moni rakastaa. Kurssin peliprojekti on pidettävä suhteellisen abstraktina mahdollisimman pitkään tai valittava siten, että yksinkertaisen esimerkin jälkeen opiskelijat voivat keskittyä omaan mielenkiinnon kohteeseensa. Tällainen abstrakti pelin alku voisi olla esimerkiksi pallopeti, joka esiintyy niin LUMA Saimaan kuin Unitynkin omissa ohjevideoissa. Opetuksen edetessä pallosta voitaisiin muokata ihmismäisempi ja upottaa kamera hahmon silmien tasolle, jolloin saadaan ensimmäisen persoonan näkökulmasta toimivan pelin runko. Tästä opiskelijat voisivat jatkaa itsenäisesti haluamaansa suuntaan, esimerkiksi lisäämällä hahmolle aseensa jolloin tuloksena olisi räiskintäpeti, tai luomalla

pelimaailmaan haasteita pelaajan ratkottavaksi pulmapelimäisesti. On kuitenkin otettava myös kurssin aikaraja huomioon — 3D-projekti vie huomattavasti enemmän aikaa kuin kaksiulotteinen. Opiskelijat odottavat luultavasti voivansa rakentaa jotain valmiin näköistä, mikä ei 30:n tunnin aikarajoitteella ole täysin omin käsin mahdollista. Esimerkkipelin viimeistelyn apuna voidaan kuitenkin hyödyntää internetistä löytyviä ilmaisia 3D-objekteja, jolloin lopputuloksesta saadaan koristellumpi.

Vaikka otos oli suhteellisen pieni, tulokset vastaavat melko tarkasti suurempien, maailmanlaajuisten kyselyiden tuloksia. Yhdysvaltain pelialan kauppaliitto ESA (Entertainment Software Association) järjestää vuosittain kyselyitä Yhdysvalloissa muun muassa pelaajien demografiasta ja heidän pelimieltymyksistään. Vuoden 2017 kyselyssä [29] suosituimmaksi monipelityylilajiksi nousi 29 %:lla äänistä räiskintä, joka oli myös tässä kyselytutkimuksessa erittäin suosittu. Toiseksi suosituin tyyli laji Yhdysvalloissa ovat niin sanotut “casual” eli rennot pelit, joihin myös monet pulmapeli-tyylilajin pelit voidaan lukea. Räiskintäpelit ovat myös 1. ja 2. sijoilla vuoden 2016 myydyimpien pelien listalla ja ne muodostivat 27.5 % kaikista myydyistä peleistä vuonna 2016.

5. KURSSIN MUOVAAMINEN KYSELYN PERUSTEELLA

Tässä luvussa muovataan Lappeenrannan teknillisen yliopiston yhdessä LUMA-keskus Saimaan kanssa tuottamia lukio-opetuksen oppimateriaaleja SVK:ssa järjestetyn kyselyn tulosten mukaisiksi. Lisäksi annetaan yleisiä suosituksia lukion ohjelmointikurssin rakenteesta ja sisällöstä kyselyn perusteella.

5.1 Yleisiä suosituksia

Viisi tärkeintä suositusta peliohjelmointikurssille ovat:

- Kurssin opiskelijoilla ei saa olettaa olevan aiempaa ohjelmointikokemusta
- Ohjelmoinnin opetukseen on käytettävä suurin osa kurssin tunteista
- Myös muita pelin tekemisen aihealueita kuten pelitaide, -suunnittelu ja -testaus tulisi, jos mahdollista, käydä läpi
- Tiedonhakua on opetettava
- Kurssin on kannustettava itsenäiseen opiskeluun ja tekemiseen

Kurssin on lähdettävä aivan alkeista. Ohjelmoinnin opetukseen on siis käytettävä suuri osa kurssin tunteista. Tämä voi muodostua ongelmaksi varsinkin kouluissa, joissa muita lukion ohjelmointikursseja ei ole tarjolla, kuten SVK:ssa. Tämän lisäksi muutama oppitunti tulisi, jos mahdollista, omistaa pelin tekemisen muille aihealueille kuten pelitaiteelle, -suunnittelulle ja -testaukselle. Lukioissa, joissa on monta pelialan kurssia, voidaan nämä aihealueet jakaa usealle kurssille.

Opiskelijoita on kannustettava opiskelemaan myös itsenäisesti, varsinkin jos aika on rajattu. Esimerkiksi kun pelisuunnittelun alkeet on käyty oppitunnilla läpi, voidaan oman pelin suunnittelu jättää kotitehtäväksi. Näin opiskelija voi itse määrittää, kuinka viimeistellyn ja monimutkaisen haluaa tehdä pelistään ja kuinka paljon aikaa käyttää sen tekemiseen. Opiskelija

pääsee erikoistumaan itselleen mielenkiintoisiin aihealueisiin, minkä myös ACM mainitsi tärkeäksi osaksi opetusta pelialan monihaarausuden takia [10].

Kurssilla tehtävän pelin tulisi olla kolmiulotteinen tietokonepeli, mieluiten kuitenkin sellainen, että siitä voi muokata haluamansalaisen 3D-pelin. Opiskelijalle annetaan siis mahdollisuus jatkaa peliä kurssin loppupuolella ja kurssin jälkeen haluamaansa suuntaan. Kurssia ja esimerkkipeliä suunniteltaessa tulisi pitää tavoitteena se, että opiskelija voi kurssin jälkeen jalostaa esimerkkipeliä eteenpäin. Kurssilla opettujien tekniikoiden tulisi riittää oman, yksinkertaisen räiskintä- ja pulmapelin luomiseen.

Tärkeää on myös painottaa opiskelijoille, että ensimmäisestä pelistä tuskin tulee Counter Striken tai muiden maailmankuulujen pelien tasoinen. Kuten Mohammed Al-Bowin et al. P4Games-kesäleiritutkimuksessa [12] selvisi, ensimmäisillä ohjelmointikerroilla on tärkeämpää poistaa epävarmuuksia ja kannustaa kokeilemaan kuin hakea täydellisiä suorituksia. On myös tärkeää opettaa opiskelijoille tiedonhakuja ja itseopiskelua. Onhan suuri osa ohjelmoinnista tiedon etsimistä. Tekniikoita ja työkaluja on olemassa tuhansia, eikä kukaan osaa niistä jokaista. Ohjelmoijan on osattava itse etsiä tai pyytää apua ongelmiinsa.

5.2 Kurssisuunnitelma LUMA:n kurssia pohjana käyttäen

Sen sijaan, että kurssi luotaisiin työtä varten täysin tyhjästä, on kurssin pohjana LUMA-keskus Saimaan peliohjelmointikurssi [4]. Ennen kyselyn tulosten huomioonottamista kurssi on rakennettu Roll-A-Ball -pelin ympärille. Peli on yksinkertainen esimerkkipeli, jossa pelaaja liikuttaa palloa, joka pyörii pelimaailmassa keräten pisteitä. Samaa esimerkkipeliä käyttää myös esimerkiksi Unity omissa esimerkkivideoissaan [16]. LUMA-kurssin koko runko on listattu taulukossa 4. Kurssi kestää 16 viikkoa, joista joka toisella on luento ja joka toisella opiskelijat tekevät harjoituksia eteenpäin oman opettajansa johdolla. Luentoja on yhteensä 7 — kaksi viimeistä viikkoa on omistettu oman pelin viimeistelylle.

Taulukko 4. LUMA-keskus Saimaan Unity-ohjelmointikurssin sisältö [4].

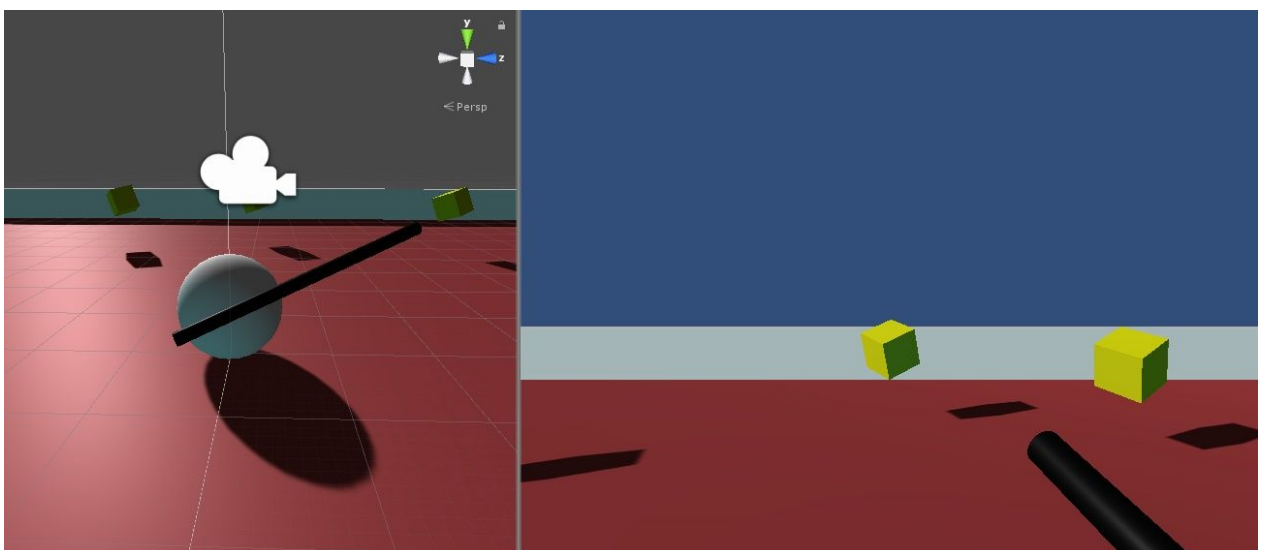
Aiheen nimi	Luennon sisältö
1. Mitä kurssilla tehdään? (Viikot 1 ja 2)	- Infoa pelialasta - Kurssin sisältö - Tietoja Unitystä
2. Ohjelmointi taitona (Viikot 3 ja 4)	- Ohjelmoinnin historiaa - Ohjelmoinnin alkeet (muuttajat) - Työvälineet - Koodin kommentointi - Roll-A-Ball -pelin ensimmäinen osa
3. Valintarakenne (Viikot 5 ja 6)	- Ohjelmoinnin perusteiden jatkoa - Valintarakenne ja muuttajien vertailu - Roll-A-Ball -pelin toinen osa
4. Toistorakenne (Viikot 7 ja 8)	- Toistorakenne - Roll-A-Ball -pelin kolmas osa
5. Tekstuurien ja äänien lisääminen (Viikot 9 ja 10)	- Tekstuurit Unityssä - Tekstuurien lisääminen Roll-A-Ball -peliin - Äänet Unityssä - Äänien lisääminen Roll-A-Ball -peliin
6. Kuulapelin kentän paranteleminen, assettien käyttäminen, aloitus- ja lopetusruudut (Viikot 11 ja 12)	- Unity Asset Storen käyttäminen - Käyttöliittymän tekeminen Unityllä - Uusien assettien lisääminen Roll-A-Ball -peliin
7. Pelintekeminen ammattina, GDD (Viikot 13 ja 14)	- Pelitestaus - Peliyritysten toiminta - Alan opiskelu - Oman pelisuunnitteludokumentin rakentaminen

Opetus alkaa kurssin rakenteen, pelialan ja Unityn esittelemisellä. Heti tämän jälkeen aloitetaan Roll-A-Ball -pelin tekeminen. Ohjelmointitaitoja opetetaan hieman jokaisen oppitunnin alussa, pelin vaatimusten mukaisesti. Esimerkiksi kolmannella luennolla esitelty valintarakenne auttaa

ymmärtämään, miksi Collider-komponentin lisääminen pelialueen reunaan saa pallon pysähtymään siihen törmätessään.

LUMA-kurssi vastaa jo melko tarkasti kyselyn tulosten luomia vaatimuksia. Se lähtee aivan alkeista ja aloittaa oman pelin koodaamisen opetuksen rinnalla heti alusta. Esimerkkipelinä on 3D-peli, kuten kyselyssä toivottiin. Kurssin loppupuolella opiskelijoille kerrotaan myös, miten ulkopuolista taidetta saa lisättyä peliin, ja miten omia pelejä suunnitellaan. Näin opiskelijalla on kaikki tarvittavat keinot jatkaa pelinkehitystä kurssin jälkeenkin, ja pelitaiteesta kiinnostuneet opiskelijat pääsevät näkemään omaa taidettaan pelissä.

Kyselyssä suosituin tyylilaji esimerkkipelille oli räiskintä. 30 tuntia on liian lyhyt aika klassisen räiskintäpelin kuten esimerkiksi Counter-Striken kopioimiseen. Esimerkkipelistä voi kuitenkin muokata räiskintämäisen siten, että Roll-A-Ball -pelin kamera siirretään pallon lähelle jolloin se näyttää olevan ihmishahmon pään korkeudella ikään kuin pelaajan silminä, kuten kuvassa 7 on esitetty. Opiskelija voi sitten lisätä pallolle aseensa sopivaan kohtaan ja ajan salliessa ohjelmoida aseensa myös ampumaan esimerkiksi hiirellä klikattaessa. Tällaisilla pienillä muutoksilla esimerkkipelistä saa nopeasti uniikin kokemuksen.



Kuva 7. Roll-A-Ball -pelistä muokattu ensimmäisen persoonan räiskintäpeli

LUMA käyttää kurssillaan CodeAcademyn Javascript-kursseja ohjelmoinnin opetukseen luentojen rinnalla. Ne kuitenkin menettävät suuren osan arvostaan Unityn hiljalleen luopuessa Javascript-tuestaan [6]. Jokin alustus ohjelmointiin kurssilla on kuitenkin oltava ennen siirtymistä Unityyn, jollaisena Javascript-kurssi voisi toki toimia. Esimerkiksi Catlike Codingin Unity-aloittelijakurssit [19] olisi kuitenkin parempi pohja kuin CodeAcademyn kurssi, sillä ne käyttävät C#:a ja lähtevät aivan alkeista, selittäen koodin jokaisen osan samalla kun sitä lisätään harjoitusprojektiin. Ne tulisi kuitenkin suomentaa ennen kurssilla käyttöä, jotta kielitaidosta ei muodostu este alkeiden opettelulle.

Ennen ohjelmoinnin aloittamista kurssilla voidaan hyödyntää Unityn omia ohjemoituja [13]. Näin opiskelija pääsee helposti ja interaktiivisesti tutustumaan Unityn käyttöliittymään ja periaatteisiin. Kun Unity-editori on päältäpäin tuttu, voidaan siirtyä ohjelmoinnin harjoitteluun.

Taulukko 5. Päivitetty kurssirakenne

Luennon nimi	Luennon sisältö
1. Mitä kurssilla tehdään?	- Infoa pelialasta - Kurssin sisältö - Tietoja Unitystä
2. Ohjelmointi taitona	- Ohjelmoinnin historiaa - Työvälineet - Unityn interaktiiviset oppitunnit - Ohjelmoinnin alkeet (muuttujat) - Koodin kommentointi - Roll-A-Ball -pelin ensimmäinen osa
3. Valintarakenne	- Ohjelmoinnin perusteiden jatkoa - Valintarakenne ja muuttujien vertailu - Roll-A-Ball -pelin toinen osa
4. Toistorakenne	- Toistorakenne

	- Roll-A-Ball -pelin kolmas osa
5. Tekstuurien ja äänien lisääminen	- Tekstuurit Unityssä - Tekstuurien lisääminen Roll-A-Ball -peliin - Äänet Unityssä - Äänien lisääminen Roll-A-Ball -peliin
6. Kuulapelin kentän paranteleminen, asettien käyttäminen, aloitus- ja lopetusruudut	- Unity Asset Storen käyttäminen - Käyttöliittymän tekeminen Unityllä - Uusien asettien lisääminen Roll-A-Ball -peliin
7. Pelintekeminen ammattina, GDD	- Pelitestausta - Peliyritysten toiminta - Alan opiskelu - Oman pelisuunnitteludokumentin rakentaminen

Taulukossa 5 on esitetty LUMA-kurssin päivitetty rakenne. Ainoa muutos alkuperäiseen rakenteeseen on Unityn interaktiivisten oppituntien sisällyttäminen alkupään oppitunteihin. Lisäksi kotitehtäviksi annetaan CodeAcademyn sijaan Catlike Codingin oppituntien tekemistä. Roll-A-Ball -esimerkkipeli on erinomainen lähtökohta, josta opiskelija voi muokata lähes minkä tahansa haluamansa pelityypin. Vaikka vastaajien keskuudessa oli suuri mielenkiinto myös muihin pelin kehityksen aihealueisiin kuten 3D-mallinnukseen, ei 30 tuntia riitä niihin keskittymiseen.

6. YHTEENVETO

Helsingin Suomalais-venäläisessä koulussa järjestettyyn kyselytutkimukseen osallistui 24 opiskelijaa. Mielipiteitä kerättiin laajalti, ja vastauksien tarkkuutta parannettiin järjestämällä kysely haastattelunomaisesti. Vastaajat olivat rohkeita antaessaan kommentteja kurssin sisältövaihtoehtoista, joten tuloksia voidaan hyvin käyttää pohjana vanhoja kursseja päivittäessä ja uusia suunniteltaessa.

Kyselyyn osallistuneiden opiskelijoiden mielipiteet vastaavat pitkälti maailmalla järjestettyjen kyselytutkimusten tuloksia. Suosituimpia tyylilajeja olivat räiskintä ja pulmapelit, jotka olivat myös Yhdysvaltojen vuoden 2016 tutkimusten kärjessä [29]. Tyylilajeista mikään ei ollut selkeästi suosituin, vaan moni aihe sai paljon sekä 0- että 5-arvosanoja. Räiskinnän keskiarvosana oli suurin, kun taas pulmapeli-tyylilaji sai vähiten huonoja arvosanoja. Suosituin pelialusta oli älypuhelin. Vastaajat olivat kiinnostuneita myös muista pelinkehityksen osa-alueista, mutta 30 tuntia on liian lyhyt aika niin laajaan opetukseen. Aiheet tulisi jakaa useammalle kurssille.

LUMA-keskus Saimaan kurssia päivitettiin tulosten perusteella. Kurssin alkuun lisättiin Unityn sisäänrakennetut oppitunnit, jotka tutustuttavat opiskelijan Unityn käyttöliittymään. Lisäksi kotitehtäväksi suositellaan annettavaksi esimerkiksi Catlike Codingin C#-oppitunteja CodeAcademyn Javascript-oppituntien sijasta, koska Unity on luopumassa Javascript-tuestaan.

Jatkotutkimuksissa olisi mielenkiintoista tehdä laaja koko Suomen kattava tutkimus maan yleisistä mielipiteistä, ja verrata niitä sitten paikkakuntien sisäiseen dataan. Voi olla, että Helsingin koulun tulokset poikkeavat suurestikin esimerkiksi Rovaniemen nuorison mielipiteistä.

7. LÄHTEET

- [1] F. Mäyrä, J. Karvinen, ja L. Ermi, ”Pelaajabarometri 2015: Lajityyppien suosio”, 2016.
- [2] ”Developer Economics Q3 2014: State of the Developer Nation”, *VisionMobile*, 18.03.2016. [Verkossa]. Saatavissa: <https://web.archive.org/web/20160318045147/http://www.visionmobile.com/product/developer-economics-q3-2014/>. [Viitattu: 24.01.2018].
- [3] Unity, ”Fast Facts”, *Unity*. [Verkossa]. Saatavissa: <https://unity3d.com/public-relations>. [Viitattu: 20.09.2017].
- [4] LUMA Saimaa, ”Kurssimateriaali”. [Verkossa]. Saatavissa: http://www2.it.lut.fi/courses/LUMA_Pelihojelmointi/library.html. [Viitattu: 05.02.2018].
- [5] J. Helminen, ”OPS2016”, *Koodiaapinen*, 10.04.2014. [Verkossa]. Saatavissa: <http://koodiaapinen.fi/ops2016/>. [Viitattu: 22.11.2017].
- [6] R. Fine, ”UnityScript’s long ride off into the sunset”, Unity Technologies Blog, 11.08.2017. [Verkossa]. Saatavissa: <https://blogs.unity3d.com/2017/08/11/unityscripts-long-ride-off-into-the-sunset/>. [Viitattu: 20.09.2017].
- [7] R. Manjula, R. Chakradhar Raju, ja M. Sai Chand, ”Software Engineering Challenges in Game Development”, *International Journal for Research in Applied Science & Engineering Technology*, vsk. 4, nro 11, s. 555-559, 2003.
- [8] J. Blow, ”Game development: Harder than you think”, *Queue*, vsk. 1, nro 10, s. 28, 2004.
- [9] U. Taipale-Lehto ja J. Vepsäläinen, ”Report on competences and skills needs in the games industry”, Opetushallitus, 4, 2016.
- [10] ACM Computing Curricula Task Force, Toim., Computer Science Curricula 2013: Curriculum Guidelines for Undergraduate Degree Programs in Computer Science. ACM, Inc, 2013.
- [11] CSTA, ”CSTA K–12 Computer Science Standards”. [Verkossa]. Saatavissa: <http://www.csteachers.org/page/standards>. [Viitattu: 21.10.2017].

- [12] M. Al-Bow et al., "Using game creation for teaching computer programming to high school students and teachers", *ACM SIGCSE Bulletin*, vsk. 41, nro 3, s. 104–108, 2009.
- [13] W. Goldstone, "Get Started in Unity with Interactive Tutorials", *Unity Technologies Blog*, 07.09.2017. [Verkossa]. Saatavissa: <https://blogs.unity3d.com/2017/09/07/get-started-in-unity-with-interactive-tutorials/>. [Viitattu: 14.10.2017].
- [14] "SimilarWeb - Digital Market Intelligence & Website Traffic", *SimilarWeb - Digital Market Intelligence & Website Traffic*. [Verkossa]. Saatavissa: <https://www.similarweb.com/>. [Viitattu: 29.01.2018].
- [15] Unity, "Learn - Modules". [Verkossa]. Saatavissa: <https://unity3d.com/learn/tutorials>. [Viitattu: 30.09.2017].
- [16] "Roll-a-ball tutorial", Unity. [Verkossa]. Saatavissa: <https://unity3d.com/learn/tutorials/projects/roll-ball-tutorial>. [Viitattu: 30.09.2017].
- [17] Catlike Coding, "About Catlike Coding". [Verkossa]. Saatavissa: <http://catlikecoding.com/about/>. [Viitattu: 30.09.2017].
- [18] Catlike Coding, "Unity C# Tutorials". [Verkossa]. Saatavissa: <http://catlikecoding.com/unity/tutorials/>. [Viitattu: 30.09.2017].
- [19] Catlike Coding, "Game Objects and Scripts, a Unity C# Basics Tutorial". [Verkossa]. Saatavissa: <http://catlikecoding.com/unity/tutorials/basics/game-objects-and-scripts/>. [Viitattu: 14.09.2017].
- [20] Noobtuts, "About". [Verkossa]. Saatavissa: <https://noobtuts.com/about>. [Viitattu: 30.09.2017].
- [21] Noobtuts, "Unity Tutorials". [Verkossa]. Saatavissa: <https://noobtuts.com/unity>. [Viitattu: 30.09.2017].
- [22] J. Horton, "Game programming for beginners", *Game Code School*. [Verkossa]. Saatavissa: <http://gamecodeschool.com/>. [Viitattu: 29.10.2017].
- [23] J. Horton, "Android Game Programming: From beginner to developer", *Game Code School*, 14.10.2016. [Verkossa]. Saatavissa: <http://gamecodeschool.com/courses/android-game-programming/>. [Viitattu: 30.09.2017].

- [24] J. Horton, "Coding a Snake Game for Android", Game Code School, 25.09.2017. [Verkossa]. Saatavissa: <http://gamecodeschool.com/android/coding-a-snake-game-for-android/>. [Viitattu: 30.09.2017].
- [25] Unity, "Company - News", Unity. [Verkossa]. Saatavissa: <https://unity3d.com/public-relations/news>. [Viitattu: 20.09.2017].
- [26] J. Smykil, "Apple Design Award winners announced", Ars Technica, 09.08.2006. [Verkossa]. Saatavissa: <https://arstechnica.com/gadgets/2006/08/4937/>. [Viitattu: 20.09.2017].
- [27] A. Robertson, "Unity officially releases its new game engine: Unity 5", The Verge, 03.03.2015. [Verkossa]. Saatavissa: <https://www.theverge.com/2015/3/3/8142099/unity-5-engine-release>. [Viitattu: 20.09.2017].
- [28] T. Sweeney, "Unreal Engine is Now Free!", 03.02.2015. [Verkossa]. Saatavissa: <https://www.unrealengine.com/en-US/blog/ue4-is-free>. [Viitattu: 20.09.2017].
- [29] "Essential facts about the computer and video game industry", ESA, 2017. Saatavissa: http://www.theesa.com/wp-content/uploads/2017/09/EF2017_Design_FinalDigital.pdf

LIITTEET

LIITE 1. Kysely lukiolaisille

1. Ikäsi?

2. Minkä tason opiskelija olet?

- Peruskoulu
- Lukio
- Ammattikoulu
- Jokin muu

3. Osaatko ohjelmoida (edes vähän)?

- Kyllä
- En

4. Jos osaat, millä ohjelmointikielellä/-kielillä?

5. Mitä ohjelmointikieliä osaat nimetä? Kirjoita kaikki jotka tiedät.

6. Montako tuntia viikossa keskimäärin pelaat/katsot/harrastat videopelejä?

- 0
- 1-2
- 3-6
- 7-12

(jatkuu)

LIITE 1. (jatkoa)

13-20

21-30

30+

7. Millä laitteilla pelaat pelejä? (Voit valita monta)

Älypuhelin

Tietokone

Konsolit

Jokin muu

8. Oletko käyttänyt Unity-pelimoottoria aiemmin?

Kyllä

En, mutta olen kuullut siitä

En ole kuullutkaan

9. Osaatko luoda 3D-malleja tietokoneella (millään tasolla)?

Kyllä

En

10. Osaatko piirtää tietokoneella (millään tasolla)?

Kyllä

En

11. Oletko suunnitellut omaa peliä (millään tasolla)?

Kyllä

En

(jatkuu)

LIITE 1. (jatkoa)

12. Oletko tarkoituksella etsinyt pelaamistasi peleistä vikoja/bugeja? (esim. peli sammuu kun teen näin)

Kyllä

En

13. Kuinka kiinnostunut olet seuraavista mahdollisista kurssin aiheista (0 = en yhtään, 5 = todella kiinnostunut)?

	0	1	2	3	4	5
Koodaaminen						
3D-mallinnus						
2D-pelitaiteen piirtäminen						
Pelitestaus						
Pelisuunnittelu						

14. Haluaisitko kurssilla tehtävän pelin olevan mieluummin 2D- vai 3D-peli?

2D

3D

15. Mille laitteelle/laitteille peli pitäisi tehdä?

Tietokone

(jatkuu)

LIITE 1. (jatkoa)

Älypuhelin

Selain (nettipeli)

Jokin muu: _____

16. Minkä tyylistä peliä haluaisit tehdä kurssilla (0 = ei kiinnosta, 5 = todella kiinnostunut)?

	0	1	2	3	4	5
Räiskintä (esim. Counter Strike)						
Pulmapeli (esim. Candy Crush, Tetris, 2048)						
Simulaatio (esim. Farming Simulator, Stardew Valley, Sims)						
Tasohyppely (esim. Mario, Shovel Knight)						
Arcade (esim. Pac-Man, Snake)						
Räiskintä (esim. Counter Strike)						
Pulmapeli (esim. Candy Crush, Tetris, 2048)						
Simulaatio (esim. Farming Simulator, Stardew Valley, Sims)						

(jatkuu)

LIITE 1. (jatkoa)

17. Mitä osia pelistä olisit kiinnostunut luomaan itse (0 = en yhtään, 5 = todella kiinnostunut)?

	0	1	2	3	4	5
2D-taide (2D-kuvat ja väritykset)						
3D-mallit						
Äänet						

18. Muita toiveita kurssille?

LIITE 2. Yhteenveto kyselyn vastauksista

1. Ikäsi?

15: 8 = 33.3%

16: 3 = 12.5%

17: 2 = 8.3%

18: 8 = 33.3%

19: 3 = 12.5%

2. Minkä tason opiskelija olet?

Peruskoulu: 9 = 37.5%

Lukio: 15 = 62.5%

3. Osaatko ohjelmoida (edes vähän)?

Kyllä: 2 = 8.3%

En: 22 = 91.7%

4. Jos osaat, millä ohjelmointikielillä/-kielillä?

- C#
- Javascript, C++
- en osaa :(
- Suomi/venäjä
- -

5. Mitä ohjelmointikieliä osaat nimetä? Kirjoita kaikki jotka tiedät.

- Java, C#, C++, JavaScript, Python, Ruby
- C++, basic, java, vala

(jatkuu)

LIITE 2. (jatkoa)

- Java Script
- JavaScript, C++, Python,
- osaan pelata gta
- ..
- -

6. Montako tuntia viikossa keskimäärin pelaat/katsot/harrastat videopelejä?

0: 6 = 25%

1-2: 8 = 33.3%

3-6: 4 = 16.7%

7-12: 2 = 8.3%

13-20: 2 = 8.3%

21-30: 0 = 0.0%

30+: 2 = 8.3%

7. Millä laitteilla pelaat pelejä? (Voit valita monta)

Älypuhelin: 15 = 62.5%

Tietokone: 10 = 41.7%

Konsolit: 8 = 33.3%

En millään: 1 = 4.2%

En pelaa: 1 = 4.2%

8. Oletko käyttänyt Unity-pelimoottoria aiemmin?

Kyllä: 5 = 20.8%

En, mutta olen kuullut siitä: 3 = 12.5%

En ole kuullutkaan: 16 = 66.7%

(jatkuu)

LIITE 2. (jatkoa)

9. Osaatko luoda 3D-malleja tietokoneella (millään tasolla)?

Kyllä: 6 = 25%

En: 18 = 75%

10. Osaatko piirtää tietokoneella (millään tasolla)?

Kyllä: 17 = 70.8%

En: 7 = 29.2%

11. Oletko suunnitellut omaa peliä (millään tasolla)?

Kyllä: 3 = 12.5%

En: 21 = 87.5%

12. Oletko tarkoituksella etsinyt pelaamistasi peleistä vikoja/bugeja? (esim. peli sammuu kun teen näin)

Kyllä: 5 = 20.8%

En: 19 = 79.2%

(jatkuu)

LIITE 2. (jatkoa)

13. Kuinka kiinnostunut olet seuraavista mahdollisista kurssin aiheista (0 = en yhtään, 5 = todella kiinnostunut)?

	0	1	2	3	4	5
Koodaaminen	3	9	3	2	1	6
3D-mallinnus	6	3	2	3	2	8
2D-pelitaiteen piirtäminen	5	4	3	1	4	7
Pelitestaus	5	2	3	3	3	8
Pelisuunnittelu	6	4	3	3	1	7
Oikeiden pelien tutkimista/havainnointia	6	3	4	4	1	6

14. Haluaisitko kurssilla tehtävän pelin olevan mieluummin 2D- vai 3D-peli?

2D: 3 = 12.5%

3D: 21 = 87.5%

15. Mille laitteelle/laitteille peli pitäisi tehdä?

Tietokone: 17 = 70.8%

Älypuhelin: 9 = 37.5%

Selain (nettipeli): 7 = 29.2%

Konsolille: 1 = 4.2%

(jatkuu)

LIITE 2. (jatkoa)

16. Minkä tyylistä peliä haluaisit tehdä kurssilla (0 = ei kiinnosta, 5 = todella kiinnostunut)?

	0	1	2	3	4	5
Räiskintä (esim. Counter Strike)	9	1	2	1	3	10
Pulmapeli (esim. Candy Crush, Tetris, 2048)	6	2	2	9	2	4
Simulaatio (esim. Farming Simulator, Stardew Valley, Sims)	6	4	1	4	4	5
Tasohyppely (esim. Mario, Shovel Knight)	7	3	2	2	6	4
Arcade (esim. Pac-Man, Snake)	8	5	2	3	4	4

17. Mitä osia pelistä olisit kiinnostunut luomaan itse (0 = en yhtään, 5 = todella kiinnostunut)?

	0	1	2	3	4	5
2D-taide (2D-kuvat ja väritykset)	8	2	2	4	8	2
3D-mallit	8	0	2	6	2	6
Äänet	8	0	2	6	4	4

18. Muita toiveita kurssille?

- Ei
- Ei mitään erikoista

(jatkuu)

LIITE 2. (jatkoa)

- Koulun counter-strike peliturnaus
- Unityn käyttöopas
- Että saisi itse tehdä paljon ja ehtisi myös testaamaan sitä peliä jonka itse tekisi
- tehkää hienoi pelei jee!!!
- -