

# Remiksi-projektin Scratch-työpaja

*Opettajan ohje*

*Tero Huttunen ja Jouni Seppänen 2015*

[Tavoite](#)

[Valmistelu](#)

[Taustatietoa](#)

[Pelisuunnittelusta](#)

[Tekijänoikeuksista](#)

[Ohjelmoinnista](#)

[Työpaja 1 \(90-135 minuuttia\)](#)

[Scratch-pelieditorin käyttöliittymään tutustuminen](#)

[Käyttäjätilin luominen Scratch-sivustolle](#)

[Editorin ominaisuuksien lyhyt esittely](#)

[Hahmojen ohjelmointi](#)

[Pelihahmon kontrollien skriptaaminen](#)

[Muiden hahmojen skriptaaminen](#)

[Pelikentän ja hahmon interaktion toteuttaminen](#)

[Loppukevennys: Scratch-pelien pelaaminen ja reflektointi sekä kotitehtävä](#)

[Työpaja 2 \(90-135 minuuttia\)](#)

[Oman hahmon ja sokkelon suunnittelu ja peliin lisääminen](#)

[Hahmon valinta kuvapankista tai piirtäminen piirtoeditorilla](#)

[Pelikentän suunnittelu](#)

[Pelin laajentaminen ja viimeistely](#)

[Pelin alku- ja loppuruutujen toteuttaminen](#)

[Äänitehosteet ja musiikki](#)

[Kuvatehosteet](#)

[Lisäkenttien ja -vaikeusasteen lisääminen](#)

[Pistelaskun toteuttaminen](#)

[Töiden esittely ja jakaminen](#)

[Oman pelin esittely](#)

[Pelin jakaminen Scratch-sivustolla](#)

[Muiden peleihin tutustuminen](#)

[Ongelmatilanteita](#)

[Oppisisällöt ja oppimistavoitteet](#)

[Tietokoneohjelmien toimintalogiikan ymmärrys](#)

[Matematiikka](#)

[Median tuottaminen ja mediakulttuuri](#)

[Pelisuunnittelu](#)

[Cc-lisenssien hyödyntäminen](#)

[Lupapyyntömalli](#)

## Tavoite

Työpajan tavoitteena on tutustuttaa osallistujat ohjelmointilogiikan alkeisiin sekä erilaisen median hyödyntämiseen digitaalisten pelien tai animaatioiden tekemisen kautta. Työpajoissa näihin osa-alueisiin tutustutaan [Scratch](#)-ohjelmiston avustuksella. Scratch on MIT:n (Massachusetts Institute of Technology) Media Labin kehittämä ohjelmisto, jonka avulla voidaan tuottaa pelejä graafisessa käyttöliittymässä ilman aiempaa osaamista ohjelmoinnista. Scratchin filosofiasta: [Scratch: Programming for All](#).

Työpaja on suunnattu ala-asteen 3.–6. -luokkalaisille, joilla ei ole aiempaa kokemusta pelien tekemisestä tai Scratch-ohjelmiston käyttämisestä. Työpajassa toteutetaan yksinkertainen sokkelopeli, joka tarjoaa mahdollisuuden sekä tutustua ohjelmoinnin perusteisiin että ilmaista itseään suunnittelemalla pelille tarina ja piirtämällä hahmot ja sokkelo. Työpajan tuloksena jokainen työpajaan osallistunut on saanut aikaiseksi oman pelin tai animaation sekä jakanut oman tuotoksensa muille käytettäväksi, jos tähän on vanhempien suostumus.

## Valmistelu

Pyydetään oppilaiden vanhemmilta lupa tehdä tunnus Scratch-sivustolle (lupapyyntömalli tämän dokumentin lopussa) ja jakaa pelejä sivustolla. Oppilaat tarvitsevat sähköposti-osoitteen, joka annetaan tunnusta luodessa ja joka pitää varmentaa: sivustolta tulee osoitteeseen viesti, jossa olevaa linkkiä pitää klikata. Virallisen ohjeen mukaan alle 13-vuotiaiden pitää käyttää vanhemman osoitetta ja 13 vuotta täyttäneet saavat käyttää omaansa. Käytetään mitä osoitetta tahansa, on hyvä tehdä tunnukset hyvissä ajoin, jotta osoitteen haltija ehtii saada viestin ja klikata linkkiä. Pelejä pystyy tekemään ilman varmennusta, mutta niiden jakaminen ei onnistu. Kirjautumisen ”tasoja” on kolme:

”kirjautumistaso”	mitä voi tehdä
ei kirjautunut, anonyymi käyttäjä	ei pysty tallentamaan sivustolle
kirjautunut, sähköpostia ei varmennettu	pystyy tallentamaan, ei pysty jakamaan eikä kommentoimaan muiden töitä
sähköpostiosoite varmennettu	pystyy jakamaan ja kommentoimaan

Joka tapauksessa on mahdollista tallentaa pelejä omalle tietokoneelle, jos tietokoneiden asetukset sen sallivat. Näin tallennettuja pelejä voi jakaa esim. USB-tikun tai jonkin pilvipalvelun kautta, mutta se on tietysti kömpelämpää kuin suoraan Scratch-sivustolla jakaminen.

Tarvitaan tietokoneluokka, jonka koneilla pystyy käyttämään kohtalaisen nykyaikaista www-selainta (esim. Chrome tai Firefox - myös Internet Explorerin versiot 10 ja 11 sopivat, mutta niiden kanssa tuntuu olevan useammin ongelmia). Selaimessa on oltava Flash-laajennus, mikä sulkee pois useimmat tablettikoneet.

Ennen työpajan ajankohtaa on hyvä seurata myös Scratch-tiimin ilmoituksia mahdollisista huoltokatkoista tai päivitysten aiheuttamista muutoksista ohjelman toiminnallisuuksiin. Hyvä

tapa seurata mahdollisia muutoksia on Scratchin Twitter-tili, osoitteessa:

<https://twitter.com/scratch>

## Taustatietoa

Työpajan ohjaajan kannattaa harjoitella yksinkertaisten pelien tekemistä Scratchilla etukäteen. Yksi hyvä suomenkielinen ohje on Nelimarkka, Vainio, Kinnunen: Ohjelmointia Scratchin kanssa, [http://avoинoppikirja.fi/tiedostot/muut/ohjelmointia\\_scratchin\\_kanssa.pdf](http://avoинoppikirja.fi/tiedostot/muut/ohjelmointia_scratchin_kanssa.pdf) Tämä opas kuvaa Scratchin vanhempaa versiota 1.4, joka piti asentaa tietokoneelle, mutta joitakin yksityiskohtia lukuunottamatta selaimessa toimiva versio 2.0 on samanlainen. Emme suosittele materiaalin luentomaista läpikäyntiä oppilaiden kanssa, mutta sen voi antaa lisäluettavaksi kiinnostuneille, ja opettajan on hyvä olla tietoinen toisto- ja ehtorakenteiden, muuttujien ja skriptien välisten viestien toiminnasta.

Ohjelmoinnin logiikan lisäksi pelejä tehdessä törmää väistämättä pelisuunnitteluun. Nykyisin digitaalisia pelejä on hyvin monenlaisia mutta miltei kaikissa niistä on samankaltaisia elementtejä tai motivaattoreita, joilla pelaaja saadaan interaktioon peliohjelman kanssa.

## Pelisuunnittelusta

[Yhden tulkinnan mukaan](#) pelien suunnittelussa tai peleissä yleensä on keskeistä seuraavat viisi elementtiä:

1. **Interaktio**, jolla tarkoitetaan sitä esimerkiksi sitä kun pelaaja saa välittömän palautteen peliohjelmasta niille toiminnoille, joita tekee esimerkiksi hiirellä tai näppäimistöllä. Mikäli interaktio ei toimi, on hidas tai epäjohdonmukainen, syntyy helposti tunne huonosta pelistä. Esimerkiksi kun pelaaja painaa hahmon ohjaamiseen tarkoitettuja nappeja ja hahmo ei joko liiku tai tekee jotain päinvastaista on peli silloin "rikki", eikä kannusta jatkamaan pelaamista. Hyvää interaktiota voidaan vahvistaa hyvien kontrollien lisäksi vaikkapa äänitehosteilla tai pelaajan hahmon animoinnilla. Kun esimerkiksi pelaaja painaa näppäintä, joka aiheuttaa pelihahmon hypyn, voidaan aktivoida samaan kohtaan käynnistyväksi hyppäämistä kuvaava lyhyt äänitehoste.
2. **Tavoite**, joka voi esimerkiksi sokkelopelissä olla aarre tai maali, jota kohti pyritään.. Vaikkakin nykyisissä hiekkalaatikkopeleissä voi seikkailla päämäärättömästi, on tällöin usein pelaaja itse, joka asettaa itselleen tavoitteita. Esimerkiksi Minecraftissa tavoitteena voi ensimmäisenä olla valmistaa työkaluja, vaikka peli itsessään ei palkitse tai rankaise vaikka ei haluaisi työkaluja valmistakaan.
3. **Kamppailu**lla pyritään tekemään tavoitteeseen pääseminen vähän haastavammaksi ja välttämään pelin muodostumista tylsäksi. Kamppailun yksinkertaisimpia symboleja voisi olla vaikkapa sokkelopelissä olevat viholliset, jotka toimivat hidasteina tavoitteeseen pääsemisessä. Esteenä toimivat hahmot voivat vaikkapa palauttaa pelaajan hahmon kentän alkuun, mikäli pelaajan hahmo koskettaa niitä. Muita kamppailun peruselementtejä peleissä ovat esimerkiksi kilpailu, jossa pelaajat pelaavat toisiaan vastaan. Kamppailu voi olla myös pelikentän läpäisy tietyn ajan kuluessa.
4. **Rakenne**, jolla tarkoitetaan niitä sääntöjä tai seiniä, joita on tehty skriptaamalla. Rakenteita voivat olla sokkelopelin seinät tai vaikkapa se tosiasia että pelaajan

hahmon ei haluta poistuvan pelikentältä. Rakenteilla voidaan fokusoida pelaajan toiminnot tietyille alueella tai tekemään tiettyjä funktioita. Toisinpäin voidaan ajatella miten sokkelopeli toimisi, mikäli pelikentässä ei olisi "painovoimaa" vaan pelihahmo alkaisi leijumaan sattumanvaraisesti ulos pelialueelta.

5. Fiktiivinen sisältö tai sisäsyntyinen merkitys on pelillinen viitekehys, jonka pelaajat hyväksyvät pelin alkaessa. Monopolia pelatessa pelaajat hyväksyvät yhdessä että paperi, jossa on merkintä "10.000" on arvokas mutta samalla tietävät että sen arvo ei säily kun pelaaja astuu lähimpään R-kioskiin. Samalla tavalla digitaalisessa pelissä voi kullaan värisistä pikseleistä muodostunut miekka olla pelaajan mielessä arvokkaampi kuin hopean värisistä tehty vastaava. Digitaalisissa peleissä voidaan toki vaikutelmaa vahvistaa rakenteilla, jotka sanovat esimerkiksi että hopean värisellä miekalla viholliselta vähenee 10 iskupistettä ja kultaisella 20, jolloin se on myös käytännössä "parempi".

Pelisuunnittelusta löytyy internetistä paljon erilaista materiaalia, joista tässä muutama maininta:

### **Introduction to Game Design**

<https://www.edx.org/course/introduction-game-design-mitx-11-126x>

### **Pelimekaniikka**

<https://wiki.metropolia.fi/display/~m0502452/Pelimekaniikka>

### **Defining game mechanics**

<http://gamestudies.org/0802/articles/sicart>

## **Tekijänoikeuksista**

Scratch-sivuston ehtoihin kuuluu, että siellä jaettu materiaali on käytettävissä cc-lisensillä "Nimeä-JaaSamoin 2.0": <https://creativecommons.org/licenses/by-sa/2.0/>

Siis valmiita pelejä saa käyttää oman pohjana tai osana, mutta niiden tekijöille pitää antaa heille kuuluva maininta.

## **Ohjelmoinnista**

Scratchin ohjelmointimalli poikkeaa jonkin verran siitä, mihin muiden ohjelmointiympäristöjen kanssa on ehkä totuttu. Jokaisella hahmolla on joukko *skriptejä*, jotka eivät vaikuta toisiinsa vaan voivat olla ajossa rinnakkain. Jotta skripti lähtee käyntiin, sen alussa pitää olla jokin käynnistyspalikka: "Kun klikataan vihreää lippua" käynnistää skriptin heti pelin alussa, muut palikat jonkin erityisen ehdon täytyessä, esimerkiksi jotain näppäintä painettaessa.

Skriptin sisällä suoritus etenee sarjallisesti, ylhäältä alaspäin. Silmukkapalikoilla kuten "ikuisesti" tai "toista 10 kertaa" suoritus saadaan palaamaan takaisin ylös. Esimerkiksi kalvon "Vastustajahahmo liikkumaan" skriptissä on silmukka, jotta hahmo liikkuu rataansa aina vain uudestaan eikä pysähdy ensimmäisen kierroksen jälkeen. Ehtopalikoilla "jos" ja "jos ... muuten" voidaan valita suoritettavaksi vain osa palikoista. Näistä esimerkiksi sopii kalvon "Ei pääse seinän läpi" skripti, joka palauttaa hahmon takaisin liikkeen alkupisteeseen, jos se koskettaa seinää. Ehtopalikoiden sisään sopivat erityiset kuusikulmaiset palikat, joilla on

totuusarvo: hahmo joko koskettaa harmaata väriä tai ei kosketa; muuttujan arvo joko on suurempi kuin 10 tai ei ole.

Skriptien väliset viestit ovat yksi lisätyökalu edelliseen: ohjelmoija voi määritellä eri viestejä, joita skriptit voivat lähettää, ja muut skriptit (esim. eri hahmoilla) voivat käynnistyä tietty viesti vastaanotettaessa. Esimerkiksi voidaan määritellä viestit ”peli alkaa”, ”pelaaja voitti” ja ”pelaaja hävisi”. Kun yksi skripti huomaa, että pelaaja voitti ja lähettää siitä viestin, kaikilla muilla hahmoilla voi aktivoitua tähän reagoivat skriptit: esiintymislavan taustakuvaksi vaihtuu onnitteluteksti, vastustajahahmo pienenee ja katoaa, jne.

Hienompi ja monimutkaisempi työkalu ovat muuttujat, jotka ovat nimettyjä muistipaikkoja, joissa voi olla numeroita tai tekstiä. Tieto-osastolla voidaan klikata ”Tee muuttuja” ja tehdä esimerkiksi ”pisteet”-niminen muuttuja, joka nollataan pelin alussa ja jota kasvatetaan aina, kun pelaaja löytää aarteen tai saavuttaa muun tavoitteen. Muuttujan arvo näkyy pelikentän yläkulmassa, mutta sen voi myös poistaa näkyvästä tai siirtää eri paikkaan. Kun muuttuja on tehty, on käytettävissä sitä vastaava palikka, joka sopii suuruusvertailupalikoiden sisään (nämä puolestaan ovat totuusarvoisia eli sopivat ehtopalikoiden sisään). Muuttujapalikka sopii myös sellaisiin palikoihin, joihin voi kirjoittaa itse numeron - esim. ”liiku 10 askelta” voisikin olla ”liiku ’vauhti’ askelta”, missä ’vauhti’ on muuttuja.

Mainittu ”Ohjelmointia Scratchin kanssa” sisältää paljon lisätietoa Scratch-ohjelmoinnista.

## Työpaja 1 (90-135 minuuttia)

Työpaja toteutetaan tietokonehuokassa siten, että jokainen oppilas työskentelee omalla tietokoneellaan ja työstää omaa versiota pelistä. Myös parityö on mahdollista, jos koneita ei riitä jokaiselle. Työpajan vaiheet ovat:

1. Scratch-pelieditorin käyttöliittymään tutustuminen
2. Hahmojen ohjelmointi
3. Pelikentän ja hahmon interaktion toteuttaminen
4. Loppukevennys: Scratch-pelien pelaaminen ja reflektointi

## Scratch-pelieditorin käyttöliittymään tutustuminen

### Käyttäjätilin luominen Scratch-sivustolle

- Esitetään Scratchin [säännöt](#).
- Jos tunnuksia ei ole vielä tehty, työpajaan osallistuva tekee tunnuksen nyt. Tätä varten työpajaan osallistuvalla on oltava tiedossaan (huoltajan) sähköpostiosoite sekä lupalappu tilin tekemiseen.

### Editorin ominaisuuksien lyhyt esittely

- Näytetään miten Scratch-sivustolla selataan pelejä (esim. sokkelopelejä: <https://scratch.mit.edu/studios/138299/>; hakusanalla ”pacman” löytää paljon Pacman-teemaisia pelejä), katsotaan valmiin pelin sisälle ja luodaan uusi peli. Aloitetaan liittymällä studioon ja remiksaamalla sokkelopelipohjaa, joka löytyy studiosta <http://ra.fi/oW6z> eli <https://scratch.mit.edu/studios/1133974/>. Näin jokainen saa pohjaksi alkeellisen sokkelon, joka on tarkoitus myöhemmin vaihtaa mielenkiintoisemmaksi.

- Esitellään pelieditorin rakenne, päätoiminnallisuudet ja skriptien rakentaminen ”palikoista”. Miten palikat koostetaan, miten irrotetaan ja kopioidaan. Editorin ominaisuudet tulevat tutuksi paremmin osatehtävien varrella.
- Näytetään, mistä on kyse hahmoissa ja taustoissa. Varsinkin taustojen löytäminen editorista voi olla hankalaa, kun oletustausta on vain iso valkoinen suorakulmio, mutta jos remiksataan sokkelopelin pohjaa, taustan ehkä huomaa paremmin.

## Hahmojen ohjelmointi

Pohjana olevassa pelissä on valmiina päähenkilönä kissa, vihollisena lepakko ja aarteena tähti. Osa oppilaista todennäköisesti keksii pian, miten nämä vaihdetaan muihin hahmoin, mutta suunnitelmassa on tutustua tällä kertaa ohjelmointiin ja tehdä hienommat hahmot itse seuraavalla kerralla. Annetaan oppilaille linkki ohjeeseen, jota voi pitää auki selaimen toisessa välilehdessä, ja varmistetaan että jokainen saa linkin auki. Ohjetta ei ole tarpeen käydä yhteisesti läpi ainakaan kovin yksityiskohtaisesti, vaan siitä voi vilauttaa kulloinkin käsiteltävää aihetta. Ajatus on, että oppilaat voivat tarkistaa asian siitä juuri silloin, kun kysymys on heille itselleen ajankohtainen.

Lyhyt linkki (esim. taululle): <http://goo.gl/xGQxHs>

Pitkä linkki: [https://docs.google.com/presentation/d/1DR1\\_HzrnAdonQaq9qWBy01NknH5H3O0zA22txhwHdiw/edit](https://docs.google.com/presentation/d/1DR1_HzrnAdonQaq9qWBy01NknH5H3O0zA22txhwHdiw/edit)

## Pelihakmon kontrollien skriptaaminen

- Tuotetaan skriptit, joilla pelaaja saa hallittua tekemäänsä hahmoa x- ja y-akseleilla. Tässä kohtaa voi olla hyvä kerrata koordinaatisto. Tähän vaiheeseen on hyvä varata riittävästi aikaa, koska skriptien idean oivaltaminen voi vaatia paljon yrittämistä. Jos joku tulee valmiiksi ennen kuin muut ovat valmiina, hänelle voi vinkata, että kalvoilla on muutakin lukemista.
- Oppilaat ehkä huomaavat, että hahmo kulkee seinien läpi. Tämä on hyvä tilaisuus mainita, että tietokone tekee vain sen, mitä on ohjelmoitu, eikä ohjelmassa ole toistaiseksi mitään tietoa seinistä. Ohjelmassa on siis bugi tai ainakin siitä puuttuu haluttu toiminto.

## Muiden hahmojen skriptaaminen

- Tuotetaan skripti, jolla vihollishahmo liikkuu automaattisesti sokkelossa. Tässä kohtaa on hyvä näyttää, miten Scratch-editorissa on kullekin hahmolle omat skriptit, joten jos vihollisen liikkeitä koodaa vahingossa päähenkilölle, väärä hahmo alkaa liikkua. Tässä tapauksessa skriptejä voi siirrellä eri hahmoille repputoiminnon avulla.
- Jos joku haluaa lisätä muita vihollishahmoja, hahmon voi kopioida leimasintyökalulla. Tällöin skriptitkin kopioituvat, joten niitä luultavasti halutaan muokata.

## Pelikentän ja hahmon interaktion toteuttaminen

- Ratkaistaan seinien läpi kävelemisen ongelma: muokataan pelihakmon kontrolliskriptiä niin, että hahmo ei kulje seinien läpi. Tempu tehdään niin, että hahmo liikkuu joka tapauksessa ensin, mutta jos se sitten koskettaa seinän väriä, se liikkuu heti takaisin. Ruudulle asti peli ei ehdi piirtää liikettä.
- Tuotetaan skripti, joka määrittelee mitä tapahtuu kun pelihakmo osuu viholliseen. Tämä tehdään pelihakmon skriptissä palikalla ”odota kunnes”, jonka sisään laitetaan

”koskettaako (hahmoa)” – huomattava, että tämä on eri palikka kuin ”koskettaako (väriä)”. Palikan jälkeen ohjelmoidaan törmäyksen seuraukset, esim. ääniefekti ja alkuun liukuminen, ja kaiken ympärille viritetään ”Ikuisesti”, jotta seuraavatkin törmäykset tunnistuvat.

- Tuotetaan skripti, joka määrittelee mitä tapahtuu kun pelihahmo pääsee aarteen luo. Voidaan vaikka animoida pieni tanssi ja lopuksi käyttää palikkaa ”pysäytä kaikki”, joka vastaa punaisen stop-nappulan painallusta.

## Loppukevennys: Scratch-pelien pelaaminen ja reflektointi sekä kotitehtävä

Oppilaita kiinnostaa pelaaminen joka tapauksessa, joten sitten kun tehtävät alkavat olla valmiita tai puhti loppuu, voidaan etsiä Scratch-sivustolta valmiita pelejä ja kokeilla niitä. Kun joku löytää hauskan pelin, pyydetään häntä kertomaan, mikä siitä tekee hyvän: hauskat kuvat, animaatiot, äänitehosteet, nopea tempo, jne. Ajatus olisi hieman ohjata oppilaita miettimään, miten pelejä on suunniteltu ja miten ideoita voisi soveltaa omiin peleihin.

Kehotetaan klikkaamaan jossain hauskassa pelissä ”Katso sisälle” ja selaamaan pelin skriptejä. Tarkoitus on näyttää, että samoista palikoista joita on itse käyttänyt voi rakentaa erilaisia lopputuloksia, ja että joissakin peleissä palikoita on hyvin paljon ja niiden kasaamiseen on tarvittu paljon suunnittelua ja työntekoa.

**Kotitehtävänä** oman pelihahmon ja ”pelimaailman” eli kentän suunnittelu paperilla ja kynällä. Tehtävänannon yhteydessä osallistujia pyydetään muistelemaan mikä tai mitkä ovat omia suosikkiahmoja ja mikä tekee niistä mieluisan itselle. Lisäksi voi pohtia, mitä asioita peleihin yleensä liittyy

## Työpaja 2 (90-135 minuuttia)

Tällä kertaa voi olla hyvä aloittaa muualta kuin tietokoneluokasta, jotta kotitehtävä saadaan purettua rauhassa. Jokaista pyydettiin suunnittelemaan oma kenttä ja hahmo. Sitten siirrytään tietokoneluokkaan, jossa on tarkoitus toteuttaa nämä Scratchin sisällä. Vaiheet ovat:

5. Oman hahmon ja sokkelon suunnittelu ja peliin lisääminen
6. Pelin laajentaminen ja viimeistely
7. Töiden esittely ja jakaminen

## Oman hahmon ja sokkelon suunnittelu ja peliin lisääminen

### Hahmon valinta kuvapankista tai piirtäminen piirtoeditorilla

- Jatketaan edellisellä kerralla aloitettua peliä, jotta tehdyt skriptit pysyvät mukana, ja korvataan siitä tausta ja hahmot.
- Tutustutaan tarkemmin Scratchin valmiisiin kuvapankkeihin, valmiin kuvan muokkaamiseen piirtoeditorissa tai piirretään hahmo alusta saakka piirtoeditorilla. Piirtoeditorin työkaluvalikon läpikäynti.

- Varsinkin, jos muokataan valmiita hahmoja, voidaan joutua piirtoeditorin vektoritilaan. Vektorihahmot ovat tarkkoja suurennettaessa, mutta vektoritila on vähän vaikeampi käyttää. Varsinkin alueiden täyttö maaliämpäri-työkalulla toimii eri tavalla kuin useimmat oppilaat luulevat. Kannattaa luultavasti käyttää bittikarttatilaa.
- Hahmolla voi olla monta *asustetta* ja esiintymislavalla voi olla monta *taustaa*. Asusteilla voi toteuttaa animaatioita ja taustoilla alku- ja loppuruutuja tai pelin tasoja.

### Pelikentän suunnittelu

- Kerrataan taas xy-koordinaatisto.
- Suunnitellaan pelikenttä, jossa pelaajan hahmo lähtee liikkeelle valitusta sijainnista x- ja y-akselilla sekä suunnitellaan aarre, matkan varrella olevat esteet tai vastustajat sekä sokkelon rakenne.
- Toteutetaan pelikenttä Scratch-esiintymislavan taustana. Tässä on hyvä huomata törmäystarkistusta varten, että seinien kannattaa olla keskenään samanvärisiä, tai enintään kahta eri väriä. Käytävien pitää olla tarpeeksi leveitä, että hahmo mahtuu kulkemaan niissä. Hahmon kokoa voi kuitenkin säätää helposti, joten jos tässä sattuu vahinko, tilanteen voi pelastaa myöhemmin.

### Pelin laajentaminen ja viimeistely

Luultavasti oppilailla on omia ideoita pelin laajentamiseksi, joten seuraavat ajatukset kannattaa ottaa ennemmin ideoina kuin valmiina käsikirjoituksena. Osa haluaa ehkä keskittyä hahmojen piirtämiseen vielä hienommiksi, osa lisätä toiminnallisuutta. Kaikkiin ideoihin liittyy ohjemateriaalissa kalvo tai useampia.

### Pelin alku- ja loppuruutujen toteuttaminen

- Pelin nimen sekä alku- ja loppuruutujen piirtäminen. Ruutujen vaihtamiseen kannattaa käyttää viestejä. Pelikentällä voi olla skripti, joka odottaa vaikkapa välilyönnin painallusta, ja lähettää sitten viestin 'peli alkaa'. Tehdään pelin hahmoille skriptit, jotka aktivoituvat "kun klikataan vihreä lippu" ja piilottavat hahmot näkyvistä, ja toisen skriptit, jotka aktivoituvat "kun vastaanotan 'peli alkaa'" ja palauttavat hahmot ja käynnistävät pelin toiminnat kuten vihollishahmon liikkeet ja törmäystarkistukset. Loppuruutuun taas voidaan siirtyä niin, että kun pelihahmon skripti huomaa voittotilanteen, se lähettää viestin 'pelaaja voitti', ja pelikentällä on tässä tilanteessa aktivoituva skripti, joka vaihtaa taustaksi onnittelukuvan.
- Mahdollisten välianimaatioiden tuottaminen

### Äänitehosteet ja musiikki

- Lisätään törmäysskripteihin ja mahdollisesti hahmon liikkumiseen äänitehosteita joko valmiista tehostepankista tai itse äänittämällä, jos koneissa on toimivat mikrofonit.
- Lisätään taustan skripteihin mahdollisen taustamusiikin soittaminen.

### Kuvatehosteet

- Hahmojen animointi asusteita vaihtamalla
- Hahmon haihtuminen olemattomiin viholliseen törmätessä



## Lisäkenttien ja -vaikeusasteen lisääminen

- Pelin ensimmäisen kentän läpäisyn jälkeisten kenttien suunnittelu ja toteutus. Mahdollisten lisävihollisten ja -aarteiden sekä vaikeusasteen kasvun suunnittelu ja toteutus. Kenttien väliin siirtymiin kannattaa käyttää viestejä, ja lisäksi kentän numero voi olla muuttujassa, joka voi vaikuttaa esim. vihollisen liikkumisnopeuteen.
- Vihollishahmo seuraamaan pelaajaa

## Pistelaskun toteuttaminen

- Pistelaskun toteuttaminen muuttujan avulla. Pisteitä voisi tulla siitä miten nopeasti sokkelon ratkaisee tai montako ylimääräistä aarretta kerää.

## Töiden esittely ja jakaminen

### Oman pelin esittely

- Esitellään lyhyesti peli muille työpajalaisille. Työpajalainen kertoo lyhyesti esimerkiksi pelin nimen, päähahmojen nimet sekä jotain kokemuksia pelin tuottamisen vaiheista tai mitä on oppinut peliä tehdessään.

### Pelin jakaminen Scratch-sivustolla

- Tämä pitää tehdä, jotta muut pääsevät katsomaan peliä. Muistetaan kirjoittaa pelin yhteyteen kiitokset kaikille, joiden grafiikoita, ääniä, musiikkia ja mahdollisesti skriptejä on käytetty apuna.
- Lisätään peli työpajan omaan studioon (Scratch-sivuston alayhteisö), jonka kautta kanssaoppilaat löytävät pelin helposti. Tämä kannattaa, koska Scratchin hakutoiminto ei päivyty aivan välittömästi.

### Muiden peleihin tutustuminen

- Tutustutaan muiden peleihin Scratch-yhteisössä sekä annetaan palautetta tai arvioita muiden peleistä positiivisessa hengessä. Peleihin voi toki tutustua myös työpajan edistyessä ennen viimeistä osatehtävää.

## Ongelmatilanteita

- Hahmo katoaa näkyvistä
  - Hahmo voi olla alueen ulkopuolella - käytä palikkaa "siirry kohtaan" ja laita koordinaateiksi 0, 0, niin hahmon pitäisi siirtyä ruudun keskelle.
  - Hahmo voi olla "piilotettu" - valitse Ulkonäkö-osastosta "näytä".
  - Hahmolla voi olla graafinen tehoste, joka tekee siitä näkymättömän - valitse "poista graafiset tehosteet".
- Hahmot käyttäytyvät oudosti
  - Jos nuolinäppäimet toimivat väärin, tarkista että x- ja y-koordinaatit eivät ole menneet sekaisin, ja että oikealle tai ylös mennessä lisätään positiivinen luku, vasemmalle tai alas mennessä lisätään negatiivinen luku.
  - Huomaa ero palikoiden "muuta x:n arvoa 10" ja "asetta x:lle arvo 0" välillä: ensimmäinen lisää x:n arvoon halutun määrän, toinen asettaa aina saman arvon.
  - Onko jokainen skripti sillä hahmolla kuin pitäisi, vai onko vihollishahmon skripti vahingossa pelaajan hahmolla? Jos näin, ota ruudun alareunasta

”Reppu” esiin, siirrä skripti sinne, vaihda oikeaan hahmoon ja siirrä skripti sille. Reppu ei oikeasti siirrä skriptejä vaan kopioi niitä, joten skripti pitää vielä poistaa alkuperäiseltä hahmolta.

- Pistelasku ei toimi
  - Kuten edellä, huomaa ero palikoiden ”muuta arvoa” ja ”aseta arvo” välillä. Pelin aluksi kannattaa nollata pisteet (”aseta arvo 0”) ja tavoitteita saavutettaessa lisätä pisteitä (”muuta arvoa 100:lla”).
- Koordinaatit ovat vaikeita tajuta
  - Lisää esiintymislavalle uusi tausta, valitse kirjastosta ”xy-grid”.
  - Liikuttele hiiren osoitinta pelialueella ja katsele sen oikeassa alanurkassa näkyviä x:n ja y:n arvoja.

## Oppisisällöt ja oppimistavoitteet

Oppisisällöt sekä oppimistavoitteet voidaan jakaa seuraaviin osa-alueisiin:

*tietokoneohjelmien toiminnan logiikan ymmärtäminen, matematiikka, median tuottaminen ja mediakulttuuri, pelisuunnittelu sekä cc-lisenssien hyödyntäminen mediatuotannossa.*

### Tietokoneohjelmien toimintalogiikan ymmärrys

- Alkeellinen ymmärrys tietokoneohjelmien toiminnasta: ohjelmat ovat ihmisen tekemiä, mitään ei tapahdu automaattisesti, vaan tietokone toimii täsmälleen ohjelman mukaan.
- Tietokoneohjelmiin liittyviä käsitteitä: sekvenssit, toisto, ehdolliset tapahtumat, muuttujat. Muutaman työpajan aikana oppilaista ei kuitenkaan kehity ohjelmoinnin asiantuntijoita, vaan käsitteiden ymmärtäminen jää luultavasti vain tämän harjoituksen kontekstiin.
- Ohjelmaideoiden kokeilu ja testaaminen (iteratiivinen kehittäminen): yleensä ensimmäinen yritys ei toimi halutusti vaan pitää kokeilla uudestaan.

### Matematiikka

- XY-koordinaatisto: hahmot sijoitetaan näytölle antamalla niiden koordinaatit ja niitä liikutetaan myös koordinaattien avulla.

### Median tuottaminen ja mediakulttuuri

- Kokemus siitä, että tietokonepelit ovat ihmisen suunnittelemaa ja tekemiä artefakteja, jotka ehkä kertovat jonkin tarinan.
- Piirtoeditorin käyttäminen, kuvan tuottaminen ja tallentaminen. Pajassa tehdään ainakin oma pelikenttä, mahdollisesti myös omat pelihahmot ellei käytetä valmiita hahmoja.
- Kuvaformaattien ominaisuuksien esittely
- CC-lisensoidun musiikin hyödyntäminen, mahdollisesti äänitehosteiden tuottaminen tai valmiiden äänitehostepankkien hyödyntäminen (Scratchin omat tai Esim. [ylen äänitehostepankki](#))
- Verkko-yhteisön toiminnan luonteen ymmärrys (Scratch-yhteisön sisällä). Miten toisten teoksia voi kommentoida, millainen kommentointi on sopivaa, millä ehdoin

mediatöitä voidaan yleensä hyödyntää.

## Pelisuunnittelu

- Pelikenttäsuunnittelu
- Hahmon ominaisuuksien suunnittelu
- Miten skriptit vaikuttavat siihen miltä esimerkiksi pelihahmolla sokkelon seinään törmääminen tuntuu, miten vaikutelmaa voi vahvistaa äänitehosteella tai hahmon animoinnilla.

## Cc-lisenssien hyödyntäminen

- Creative Commons -lisenssien hyödyntäminen ja tekijänoikeudet mediatuotannoissa. Tekijänoikeus on monimutkainen lain osa-alue, mutta perusidean omasta teoksesta määräämisestä pystyy kuka tahansa ymmärtämään. Jos Scratch-töitä jaetaan, edellytetään avoimen cc-lisenssin käyttämistä, jolloin vastaavalla lisenssillä levitettyjä töitä voi käyttää, kunhan seuraa lisenssin ehtoja.

## Lupapyyntömalli

Hyvät huoltajat!

Lapsenne on osallistumassa peliohjelmointityöpajaan **<lisää paikka, aika, järjestäjä>**.

Työpajassa käytetään Internet-sovellusta nimeltä Scratch, ja tarvitsemme lupaanne tehdäksenne lapsellenne käyttäjätunnuksen. Scratch on yhdysvaltalaisen MIT-yliopiston kehittämä lapsille suunnattu ohjelmointiympäristö. Sillä voi tehdä animaatioita, pelejä ja visuaalisia tarinoita ja jakaa niitä Internetissä. Lisätietoa voi hakea mm. osoitteista

<http://scratch.mit.edu> (virallinen sivusto)

<http://goo.gl/gRYpQk> (lehtiartikkeli Scratchin tavoitteista)

Scratchin nykyinen versio (toisin kuin yllämainitussa artikkelissa kuvattu) on Internet-selaimessa Flash-tekniikalla toimiva sovellus. Siksi sitä voi käyttää useimmilla nykyisillä tietokoneilla, ei kuitenkaan tableteilla tai älypuhelimilla.

Jotta Scratch-sovelluksella ohjelmointi sujuu kätevästi, siihen kannattaa luoda oma käyttäjätunnus, jolla pääsee näkemään omat pelinsä ja osallistumaan sivuston keskusteluihin. Tietosuojasyistä pyydämme tunnuksen luomiseen huoltajan lupaa. Tunnus ei ole aivan välttämätön kerhoon osallistumiseksi, mutta se helpottaa käytännön järjestelyjä huomattavasti. Tunnusta luodessa kysytään seuraavat tiedot:

- haluttu käyttäjätunnus; ei oma nimi vaan nimimerkki
- haluttu salasana; ei sama kuin koulun koneilla, kannattaa kirjoittaa muistiin
- syntymäkuukausi ja -vuosi sekä sukupuoli MIT:n tilastoja varten, eivät tule näkyviin minnekään
- kotimaa, joka tulee näkyviin sivustolle ja MIT:n tilastoja varten

- sähköpostiosoite, jota tarvitaan sekä unohtuneen salasanan vaihtamista varten että henkilöllisyyden varmistamiseksi – ilman toimivaa osoitetta pelejä ei voi jakaa

Tiedot rekisteröi Scratchia pyörittävä MIT. Työpajan puolesta mitään rekisteriä ei pidetä.

Sivuston käytösäännöt näkyvät osoitteessa [http://scratch.mit.edu/community\\_guidelines/](http://scratch.mit.edu/community_guidelines/)

Terveisin työpajan järjestäjät

**<järjestäjän yhteystiedot>**

## PELIOHJELMOINTITYÖPAJA

Lapsen nimi:

\_\_\_\_\_

Huoltajan nimi:

\_\_\_\_\_

Lapsen tai huoltajan sähköpostiosoite:

\_\_\_\_\_

- Lapselle saa tehdä tunnuksen scratch.mit.edu-sivustolle
- Lapsella on jo tunnus jota halutaan käyttää, tunnus ja salasana mukaan
- Lapselle ei saa tehdä tunnusta

Huoltajan allekirjoitus:

\_\_\_\_\_

Lähetätkää tämä lappu lapsen mukana ensimmäiseen työpajaan.



© 2015 Tero Huttunen ja Jouni Seppänen

*Remiksi-projektin Scratch-työpaja, Opettajan ohje* by Tero Huttunen & Jouni Seppänen is licensed under a [Creative Commons Attribution 4.0 International License](https://creativecommons.org/licenses/by/4.0/).