



# Innostusta ja oppimista: Pelinkehityksen opettaminen Game Maker-ohjelmistolla

Matti Välke

2024 Laurea



Laurea-ammattikorkeakoulu

# Innostusta ja oppimista: Pelinkehityksen opettaminen Game Maker-ohjelmistolla

Matti Välke  
Tradenomi (AMK)  
Opinnäytetyö  
Joulukuu, 2024

Matti Välke

**Innostusta ja oppimista: Pelinkehityksen opettaminen Game Maker -ohjelmistolla**

Vuosi

2024

Sivumäärä

25

---

Tämän opinnäytetyön lopputuloksena syntyy opetusmateriaali, joka selkeyttää pelinkehityksen perusteita ja tekee aiheesta helposti lähestyttävän aloittelijoille. Materiaalin tavoitteena on opettaa pelinkehityksen eri vaiheita käytännönläheisesti ja lisätä innostusta pelien tekemiseen. Opetusmateriaali hyödyntää visual scripting -menetelmää, eli no-code-lähestymistapaa, joka mahdollistaa pelien rakentamisen ilman syvällistä ohjelmointiosaamista.

Opinnäytetyössä esitetään, miten yksinkertainen peli rakentuu ideasta pelattavaksi peliksi. Keskeisenä tavoitteena on madaltaa kynnystä pelinkehitykseen tarjoamalla selkeä ja helposti ymmärrettävä esimerkki, jonka avulla pelinkehityksen eri osa-alueita on helppo oppia ja soveltaa käytännössä.

Asiasanat: game maker, oppiminen, pelinkehitys

**Laurea University of Applied Sciences**

**Abstract**

Degree Programme in Business Information Technology

Bachelor of Business Administration

Matti Välke

**Inspiration and Learning: Teaching Game Development with Game Maker**

Year

2024

Pages

25

---

The result of this thesis is educational material that clarifies the basics of game development and makes the subject easily approachable for beginners. The goal of the material is to teach the different stages of game development in a practical way and to inspire enthusiasm for creating games. The educational material utilizes the visual scripting method, also known as the no-code approach, which enables building games without in-depth programming skills.

The thesis demonstrates how a simple game is developed from an idea into a playable game. The main objective is to lower the threshold for game development by providing a clear and easily understandable example that makes it easy to learn and apply various aspects of game development in practice.

Keywords: game maker, game development, learning

## Sisällys

1	Johdanto .....	6
2	Ohjelmistot .....	7
2.1	Game Maker .....	7
2.2	Game Maker Language (GML) ja Game Maker Visual .....	8
2.3	Game Makerin käyttäminen .....	8
3	Opinnäytetyön projektin rajaukset .....	9
4	Pelimekaniikan suunnitelma ja toteutus .....	9
4.1	Pelisuunnittelu .....	9
4.2	Ohjausjärjestelmä ja kenttä .....	12
4.3	Kerättävien kohteiden logiikka .....	18
5	Konseptin testaaminen käytännössä .....	21
6	Pohdinta .....	22
	Taulukot .....	25

## 1 Johdanto

Tämän opinnäytetyön tavoitteena on luoda konsepti, joka auttaa opettamaan nuorille pelinkehitystä Game Maker -ohjelmiston avulla. Pääpaino on visual scripting -menetelmässä, eli no-code-työskentelyssä. Aihe valittiin, koska pelinkehityksen ja ohjelmoinnin opetuksessa on kasvava tarve helppokäyttöisille työkaluille. Tällaiset työkalut tekevät pelinkehityksestä saatutavaa myös niille, joilla ei ole syvällistä ohjelmointiosaamista. Työssä ei keskitytä perinteiseen tutkimukseen, vaan oppimisen innostamiseen ja oppimismateriaalin luomiseen. Materiaalin tavoitteena on madaltaa kynnystä pelinkehityksen eri vaiheiden oppimiseen.

Game Maker on tähän tarkoitukseen hyvä työkalu. Sen käyttöliittymä on intuitiivinen ja ominaisuudet monipuoliset. Opinnäytetyön tavoitteena on luoda vaiheittainen opetusmateriaali, joka opettaa pelinkehityksen perusteet alusta loppuun. Lopputuloksena syntyy ohjeistus projektipelin tekemiseen, jota voidaan hyödyntää opetuksessa. Opetusmateriaalissa käytetään Game Makerin Visual-ohjelmointitapaa.

Miksi Game Maker? Game Maker valittiin tämän opinnäytetyön pelimoottoriksi, koska opinnäytetyöntekijällä on suhteellisen pitkä kokemus ohjelmistosta, aina sen versiosta 1.4 lähtien. Pelien tekemisen opinnäytetyöntekijä on aloittanut jo lapsuudessaan Klik & Play ohjelmiston parissa. Vaikka Game Maker on kehittynyt ja muuttunut paljon vuosien aikana, sen käyttöliittymä on säilynyt yksinkertaisena ja helposti lähestyttävänä. Lisäksi Game Maker toimii myös vähän vanhemmilla tietokoneilla. Ohjelmisto sopii erityisesti 2D-pelien tekemiseen, vaikka sillä voi pystyä myös tekemään 3D-pelejä. Kuitenkin 3D-pelien kehitykseen suosittelen muita pelimoottoreita, kuten Unityä, Unreal Engineä tai vapaan lähdekoodin Godotia.

Tämän opinnäytetyön avulla pyritään madaltamaan pelinkehityksen oppimiskynnystä ja luomaan helposti lähestyttävä kokonaisuus täysin uusille kehittäjille. Game Makerin helppous, selkeys ja visuaalinen lähestymistapa tekevät siitä erinomaisen työkalun niin aloittelijoille kuin edistyneemmille käyttäjille.

Pelikehitys on harrastus, joka yhdistää tekniset taidot ja mielikuvituksen. Se opettaa ohjelmointia, ongelmanratkaisua, yhteistyötä ja luovuutta. Oulun yliopiston tutkimus (2024) osoittaa, että pelikehitys lisää opiskelijoiden motivaatiota ja auttaa oppimaan ohjelmoinnin perusteita. Kun oppilaat saavat luoda jotain omaa ja toimivaa, heidän kiinnostuksensa teknologiaa kohtaan kasvaa.

Jyväskylän yliopiston tutkimus (2024) kertoo, että pelit tukevat oppimista tehokkaasti. Pelikehitys opettaa lapsia ratkomaan ongelmia ja pilkkomaan tehtäviä pienempiin osiin. Näin he

oppivat ajattelemaan loogisesti ja hahmottamaan laajoja kokonaisuuksia. Tämä vahvistaa heidän taitojaan monilla eri elämäalueilla.

Pelikehitys auttaa myös ymmärtämään, miten ohjelmointia voi hyödyntää tulevaisuudessa. Yleisradion artikkelissa (2024) kerrotaan, että pelit kehittävät erityisesti ongelmanratkaisutaitoja. Nämä taidot ovat tärkeitä tulevaisuuden työelämässä, jossa teknologialla on suuri rooli. Pelikehityksen avulla lapset voivat kokea onnistumisen iloa ja samalla oppia koodausta ja suunnittelua.

Tutkimusten mukaan pelikehitys on matalan kynnyksen tapa oppia ohjelmointia ja luovaa ajattelua. Se ei vaadi syvällistä teknistä osaamista, vaan oppiminen tapahtuu hauskan tekemisen kautta. Tällainen lähestymistapa tekee oppimisesta helpompaa ja innostavampaa. Samalla se näyttää lapsille, että heidän omat ideansa ja taitonsa voivat muokata teknologian maailmaa. Tämä liittyy myös opinnäytetyöni tavoitteeseen, jossa haluan tehdä pelinkehityksestä helposti lähestyttävää ja inspiroivaa Game Maker -ohjelmiston avulla.

## 2 Ohjelmistot

Pelinkehityksessä oikean ohjelmiston valinta on tärkeää. Se vaikuttaa oppimiseen ja käyttökokemukseen. Hyvä ohjelmisto tekee pelien kehittämisestä helpommin lähestyttävää. Se tukee aloittelijoita, mutta tarjoaa tarpeeksi työkaluja myös kokeneille käyttäjille.

Monia ohjelmistoja pelinkehitykseen on tarjolla. Unity, Godot sekä Unreal Engine ovat suosittuja vaihtoehtoja. Tämä työ kuitenkin keskittyy Game Makeriin. Se on erityisen helppokäyttöinen ja soveltuu hyvin aloittelijoille. Game Makerilla voi luoda pelejä ilman syvällistä ohjelmointitaitoa, mikä tekee siitä loistavan työkalun opetukseen.

### 2.1 Game Maker

Game Maker on YoYo Gamesin kehittämä ohjelma erityisesti 2D-pelien tekemiseen. Sitä käyttävät niin indie-kehittäjät, opettajat kuin ammattilaisetkin. Ohjelmasta löytyy ilmainen versio sekä maksulliset Professional- ja Enterprise-versiot. Professional sopii kaupallisiin projekteihin yksityisille, kun taas Enterprise mahdollistaa pelien julkaisun konsoleille, kuten Xboxille ja PlayStationille ja se on myös tarkoitettu yrityskäyttöön. Game Maker tukee sekä visuaalista skriptausta (Visual) sekä GML-kieltä (GameMaker Language). Tämä tekee siitä sopivan kaiken tasoisille käyttäjille. (GameMaker 2024)

Game Makerilla on tehty useita tunnettuja pelejä. Esimerkiksi *Hyper Light Drifter*, *Undertale* ja *Katana ZERO* on luotu tällä ohjelmalla. Se tarjoaa monipuoliset ohjelmoinnin työkalut, jotka tekevät siitä erityisen houkuttelevan aloittelijoille, ammattilaisille sekä harrastelijoille. Ohjelmistosta on tarjolla ilmaisversio, joten sitä voidaan käyttää myös helposti pelinkehityksen opetuksessa.

## 2.2 Game Maker Language (GML) ja Game Maker Visual

Game Maker Language (GML) on ohjelmointikieli, jota käytetään Game Maker Studiossa. Sen avulla kehittäjä voi hallita tarkasti pelin logiikkaa, animaatioita, fysiikkaa ja muita yksityiskohtia. GML:n rakenne muistuttaa monia muita ohjelmointikieliä, joten kokeneet koodaajat oppivat sen nopeasti. (YoYo Games, 2023)

Game Maker Visual taas on no-code-menetelmä, jossa toiminnallisuuksia luodaan siirtämällä ja yhdistelemällä visuaalisia lohkoja. Tämä graafinen käyttöliittymä on suunniteltu erityisesti aloittelijoille. Se tekee pelinkehityksen oppimisesta helpompaa ja vähentää koodauksen aiheuttamia haasteita.

GML ja Visual täydentävät toisiaan hyvin. Kokeneet kehittäjät voivat hyödyntää GML:n joustavuutta monimutkaisempien toimintojen luomisessa. Visual taas on erinomainen työkalu opetukseen ja aloittelevien kehittäjien ensimmäisiin pelinkehitysprojekteihin.

## 2.3 Game Makerin käyttäminen

Game Makerin käyttö alkaa ohjelman lataamisella viralliselta GameMaker.io-sivustolta. Käyttäjän pitää valita sopiva versio: ilmainen, Professional tai Enterprise. Tässä opinnäytetyössä käytetään ilmaista versiota. Rekisteröitymisen jälkeen latauslinkki tulee näkyviin, ja ohjelman asennus onnistuu helposti ohjeita seuraamalla.

Käytön aloitus tapahtuu uuden projektin luomisella. Ensimmäisenä asetetaan perusasetukset, kuten pelialusta ja grafiikka-asetukset. Pelin toiminnallisuksia voi rakentaa joko visuaalisen skriptauksen tai GML-kielen avulla. Tämä tekee ohjelmasta joustavan vaihtoehdon kaiken tasoille käyttäjille.

Projektin rakentaminen etenee vaiheittain. Hahmoja, liikkeitä ja vuorovaikutuksia lisätään visuaalisia työkaluja ja koodia yhdistämällä. Ohjelma tukee myös resurssien, kuten grafiikoiden, animaatioiden ja äänien tuontia. Lisäksi Game Maker tarjoaa työkalut pelien testaamiseen ja julkaisemiseen eri alustoilla. Näiden ominaisuuksien ansiosta se on kokonaisvaltainen työkalu pelinkehitykseen.

### 3 Opinnäytetyön projektin rajaukset

Tämä opinnäytetyöprojekti on rajattu selkeään ja hallittavaan kokonaisuuteen. Siinä tehdään yksinkertainen 2D-peli käyttäen Game Maker -ohjelmiston visual scripting -ominaisuuksia. Pelin tarkoitus on tarjota oppijoille helppo ja innostava tapa tutustua pelinkehitykseen. Pääpaino on liikkumisessa ja keräilyelementeissä, eikä syvällistä ohjelmointitaitoa tarvita. Projekti keskittyy vain perustoimintoihin, kuten hahmon liikkumiseen ja kolikoiden keräämiseen.

Monimutkaisempia toimintoja, kuten vihollisten tekoälyä tai eri tasoja, ei sisällytetä. Tämä rajausta pitää projektin yksinkertaisena ja helposti toteutettavana. Se myös varmistaa, että peli voidaan tehdä nopeasti ja tehokkaasti. Näin oppimiskokemus säilyy hauskana, motivoivana ja aloittelijoille sopivana.

### 4 Pelimekaniikan suunnitelma ja toteutus

Tässä kappaleessa luodaan yksinkertainen ja nopeasti toteutettava peli käyttäen hyödyksi Game Makerin visual scripting -kieltä eli Game Maker Visualia. Pelimekaniikan suunnittelu aloitetaan hahmottamalla pelin perusidea ja tavoitteet. Peli keskittyy yksinkertaiseen toimintaan, kuten hahmon liikkumiseen ja kerättävien objektien keräämiseen. Tarkoituksena on tehdä projekti, joka on helposti toteutettavissa ja soveltuu aloittelijoille.

Suunnittelussa otetaan huomioon pelin kulku ja käyttäjän kokemus. Pelaajan tulee ymmärtää pelin säännöt nopeasti ilman monimutkaisia ohjeita. Peliin lisätään liikkumislogiikkaa, yksinkertaisia kerättäviä kohteita ja visuaalisia elementtejä, jotka tekevät siitä houkuttelevan. Kaikki tämä toteutetaan käyttämällä visuaalista logiikkaa, joka ei vaadi ohjelmointikoodin kirjoittamista. Näin pelintekijän ei tarvitse hallita ohjelmointikieliä, mikä madaltaa aloittamisen kynnystä.

Game Maker Visualin avulla voidaan määritellä hahmon liikkeet, vuorovaikutukset ja pelin toiminnallisuudet vetämällä ja pudottamalla lohkoja. Tämä tekee kehitysprosessista selkeän ja intuitiivisen. Tavoitteena on näyttää, miten yksinkertainen peli rakennetaan alusta loppuun, ja tarjota malli, jonka avulla oppijat voivat itse kokeilla ja laajentaa oppimaansa.

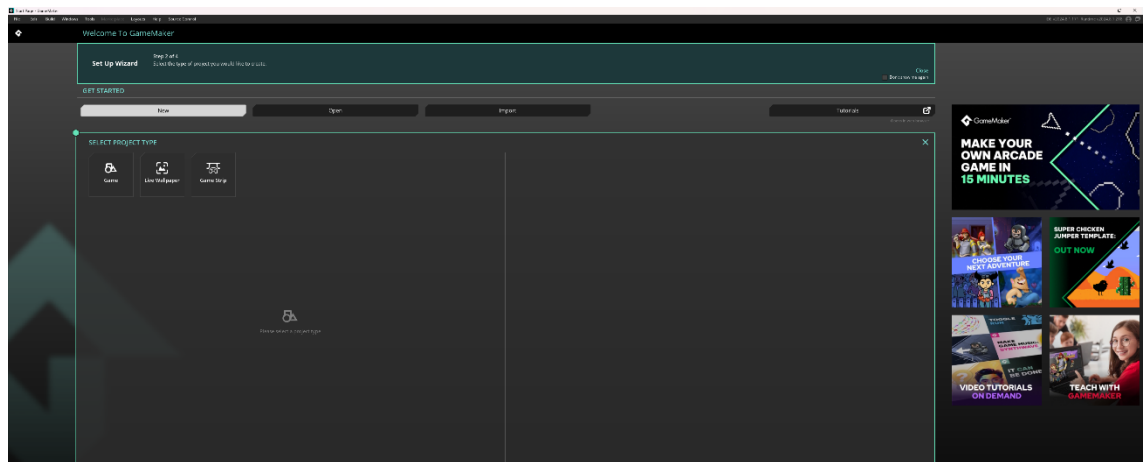
#### 4.1 Pelisuunnittelu

Luodaan 2D peli, jossa pystytään liikkumaan eri suuntiin sekä keräämään pisteitä.

Mitä tarvitsemme hyvin yksinkertaiseen peliin?

- Pelattavan hahmo
  - o Liikuu käyttäjän määrittelemistä näppäinkomennoista eri suuntiin
- Kerättävät objektit
  - o Pelaaja pystyy keräämään nämä

Aloitetaan luomalla uusi projekti:



Kuva 1. Kuvakaappaus projektin luonnin näkymästä

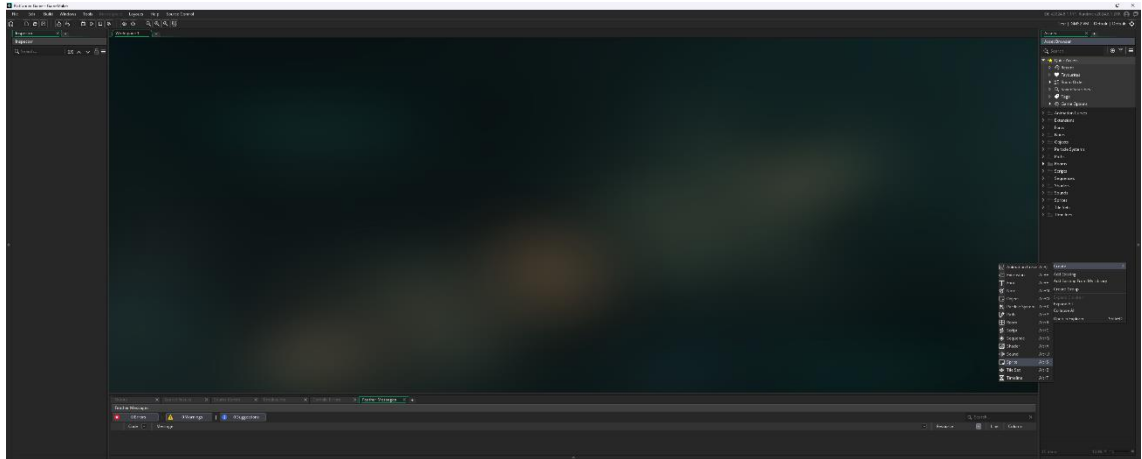
Eri projektityypit (Project Type) antavat mahdollisuuksia aloittaa jo valmiista pohjasta. Kuva 1 havainnollistaa miltä projektin luominen näyttää pelimoottorissa. Game Makerin kotisivuilta löytyy myös kattava kokoelma erilaisia tutoriaaleja, joilla voi rakentaa vastaavat pelit, joiden valmiit mallit löytyvät projektin luonnin yhteydessä avautuvasta välilehdestä.

Opinnäytetyön vuoksi aloitamme kuitenkin "Blank Gamella" eli tyhjällä pohjalla.



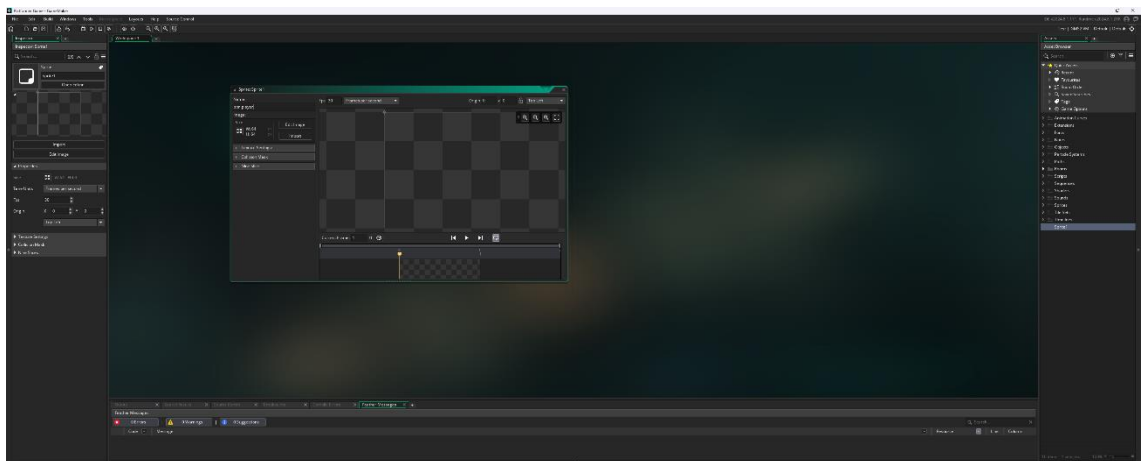
## 4.2 Ohjausjärjestelmä ja kenttä

Tässä kappaleessa keskitytään peliohjauksen yksityiskohtiin ja siihen, kuinka hahmoa voidaan ohjata. Kuva 4 havainnollistaa, miltä ensimmäisen spriten luominen näyttää pelimoottorissa. Luodaan heti ensimmäiseksi pelattavalle hahmolle sprite sekä objekti. Näissä voi käyttää myös näppäinpikakomentoja, kuten “ALT + S” spriten luomiseen ja “ALT + O” objektin luomiseen.



Kuva 4. Kuvakaappaus ensimmäisen spriten luomisesta

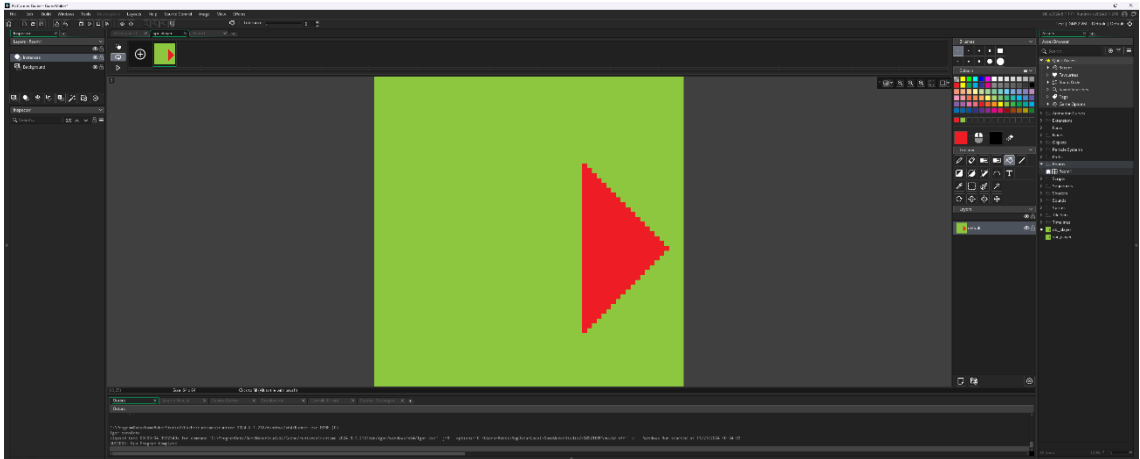
Game Makerissa ei ole vakiintuneita nimeämiskäytäntöjä. Hyvänä nyrkkisääntönä on kuitenkin nimetä tyyppi, jota seuraa alaviiva ja sen jälkeen kuvaava nimi. Käytetään pelaavan spritelle tässä tapauksessa “spr\_player”, jonka jälkeen voidaan luoda väliaikainen kuva pelihahmolle klikkaamalla “Edit Image”.



Kuva 5. Kuvakaappaus sprite editorista

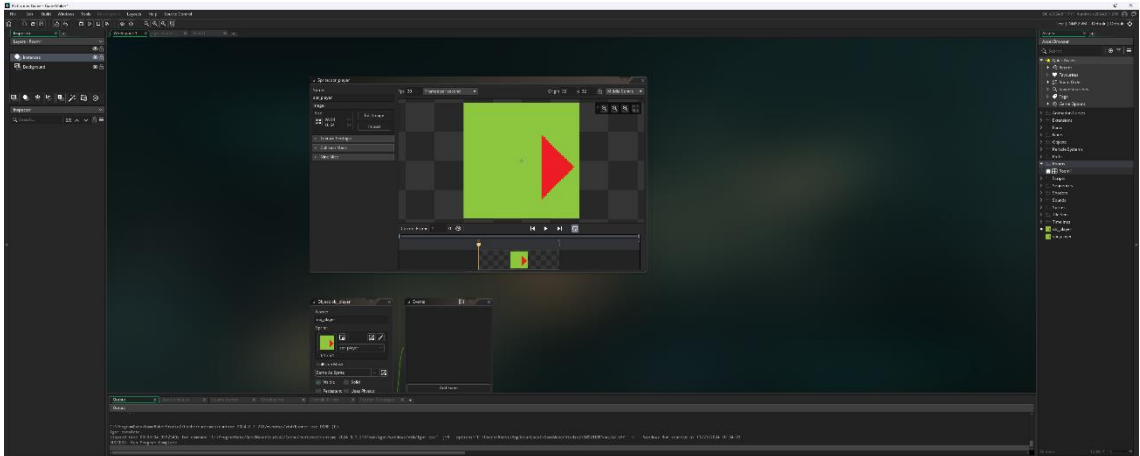
Päästään ”Sprite Editoriin” jossa voimme muokata kuvia ohjelmiston sisällä. Kuva 5 havainnollistaa sprite editorin käytön pelihahmon luomisen yhteydessä. Klikkaa sopiva väri, tässä

esimerkissä valitsimme vihreän. Valitse ”Fill Tool” joko klikkaamalla maalipurkin kuvaa tai painamalla pikanäppäintä ”F”, klikkaa tyhjää aluetta kuvassa, jonka jälkeen kuva maalataan. Luodaan kuvaan vielä suuntaa osoittava merkki. Tässä tapauksessa loimme Polygon Tool (pikanäppäin ”P”) kolmion, mutta voit käyttää esimerkiksi kynää ”pikanäppäin ”D”) luodaksesi suuntamerkin. Kuva 6 havainnollistaa sprite editorilla luodun hahmon lopullista ulkonäköä.



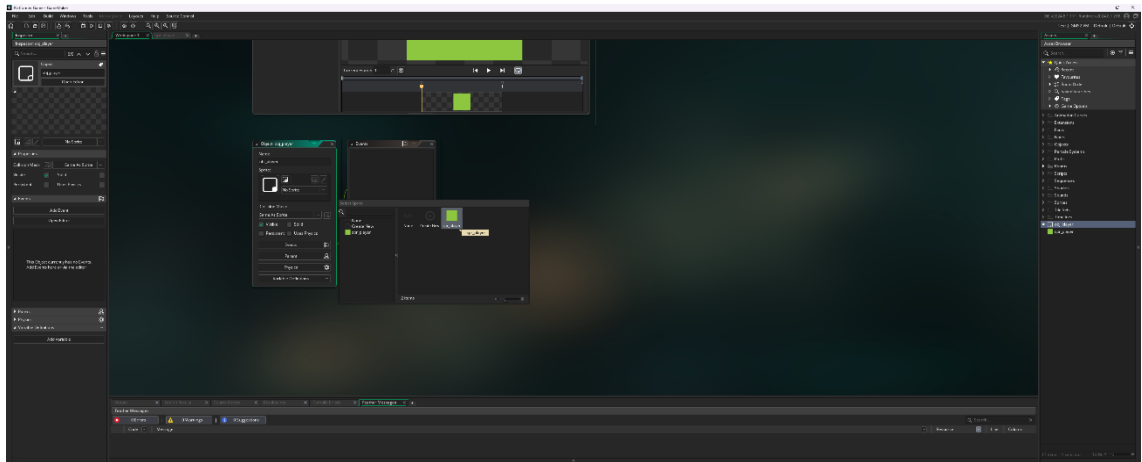
Kuva 6. Kuvakaappaus sprite editorilla tehdystä hahmosta

Voimme palata takaisin työtilaan valitsemalla välilehdeltä Workspace 1. Valitaan kuvalle vielä ”Origin” piste valitsemalla ”Middle Center”. Seuraavaksi luomme objektin hahmolle. Palaa taas sivuluetteloon ja valitse Create -> Object tai käytä näppäinyhdistelmään ”ALT + O”.



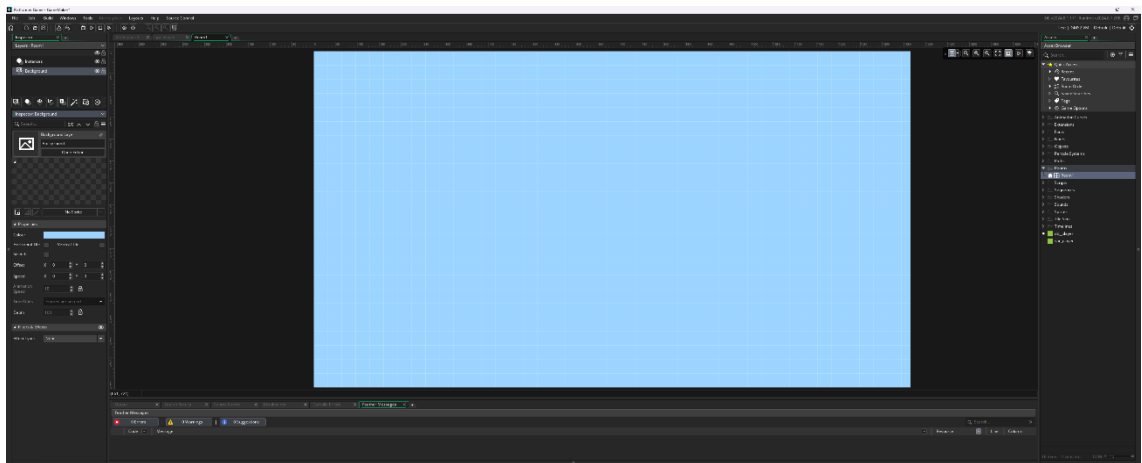
Kuva 7. Kuvakaappaus objektin luomisesta pelihahmolle

Nimetään objekti nimellä ”obj\_player” ja valitaan objektille juuri luotu ”spr\_player”. Kuvat 7 ja 8 havainnollistavat objektin luomista ja spriten liittämistä siihen. Klikkaa kohdan ”No Sprite” vierestä kolmen pisteen takaa valikko auki ja valitse ”spr\_player”.



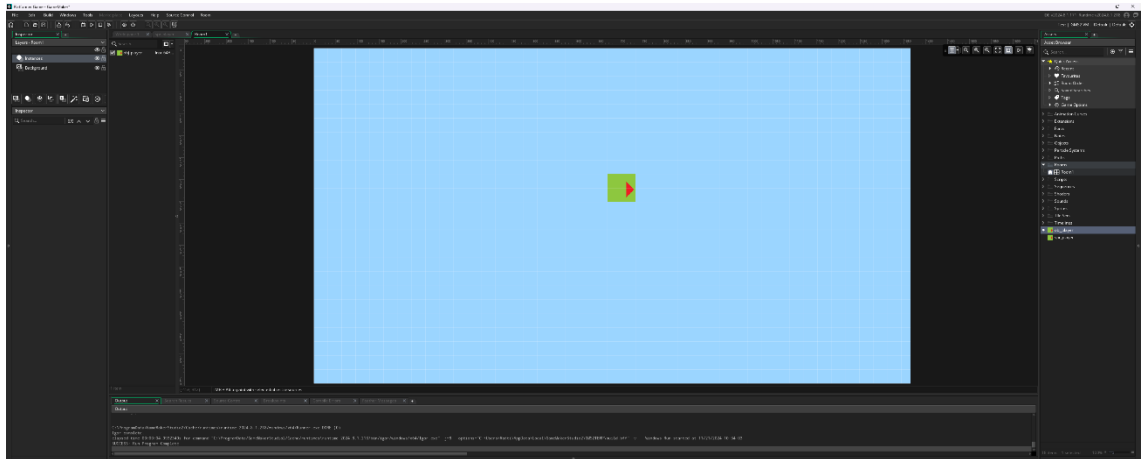
Kuva 8. Kuvakaappaus spriten lisäämisestä pelihahmon objektiin

Siirretään objekti seuraavaksi huoneeseen. Valitse oikealla olevasta valikosta auki "Rooms", jonka alta löytyy "Room1". Tuplaklikkaa päästäksesi uudelle välilehdelle. Kuva 9 havainnollistaa huone-editorin käyttöä ja taustaväriin muuttamista. Vaihdetaan musta tausta vasemmasta valikosta valitsemalla tausta, eli "Background", ja uudesta valikosta klikkaamalla "Colour". Esimerkin vuoksi käytämme sinistä, jonka HEX-arvo on 9BD5FF.



Kuva 9. Kuvakaappaus huone-editorista ja taustaväriin muuttamisesta

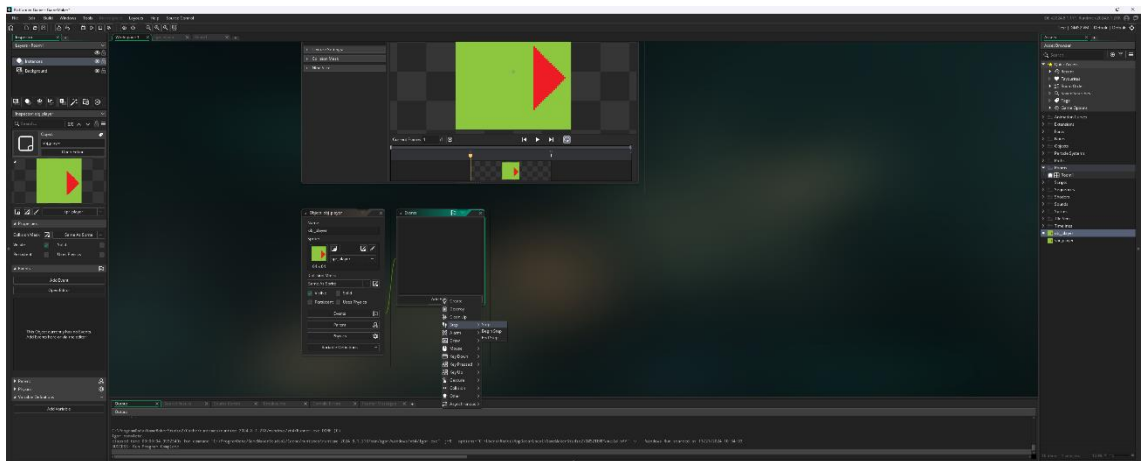
Valitaan vasemmasta välilehdestä tasojen "Layer" alapuolelta "Instances" ja raahataan oikeasta valikosta pelihahmo "obj\_player". Kuva 10 havainnollistaa, kuinka pelihahmon objekti on siirretty huoneeseen. Voimme nyt testata miltä peli näyttää klikkaamalla ylävalikosta kohtaa "Play" tai pikanäppäimellä F5. Hahmolle ei kuitenkaan ole luotu vielä mitään logiikkaa, joten mitään ei tapahdu.



Kuva 10. Kuvakaappaus jossa pelihahmon objekti on siirretty huoneeseen

Luodaan seuraavaksi hahmolle liikkumislogiikka. Pelihahmon pitää liikkua ylöspäin osoittavasta nuolinäppäimestä ja sen tulisi vaihtaa suuntaa, kun painetaan nuolinäppäintä joko vasemmalla tai oikealla.

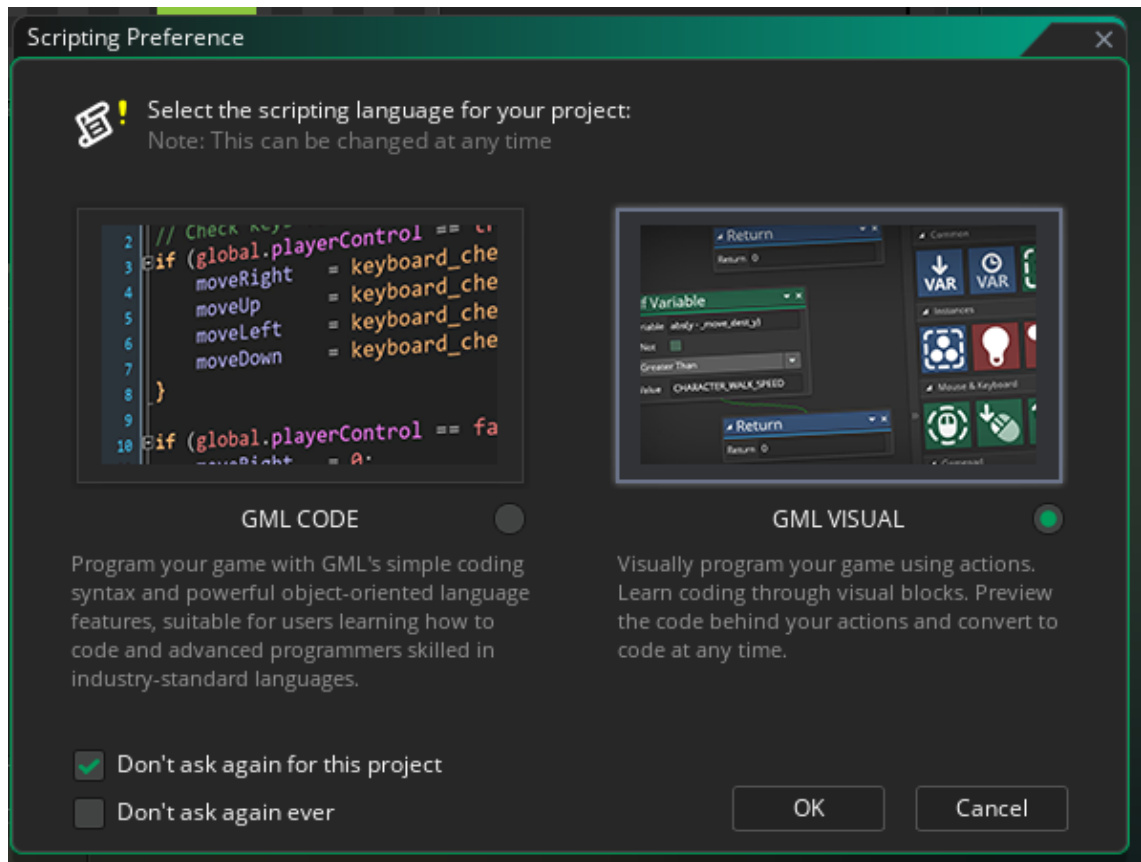
Palataan taas työtilaan (Workspace 1) ja valitaan oikealta pelihahmon objekti "obj\_player" tuplaklikkaamalla. Kuva 11 havainnollistaa ensimmäisen actionin lisäämistä pelihahmon objektiin. Klikataan "Add Event" -> Step -> Step.



Kuva 11. Kuvakaappaus ensimmäisen actionin lisäämisestä pelihahmon objektiin

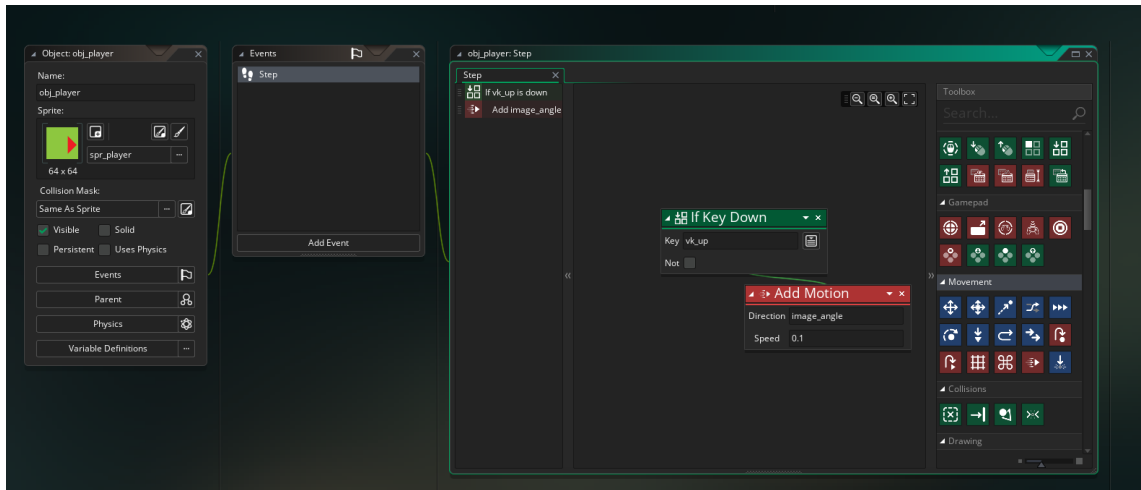
Ohjelmisto kysyy seuraavaksi, haluatko käyttää GML Codea vai GML Visualia. Kuva 12 havainnollistaa scripting-preferenssien valinnan. Tässä työssä käytämme GML Visualia, eli klikkaa

GML Visual ja lisää rasti kohtaan “Don’t ask again for this project”, niin ohjelmisto ei kysy enää tässä projektissa kumpaa haluat käyttää.



Kuva 12. Kuvakaappaus Scripting preferensseistä

Voimme nyt lisätä objektiin logiikkaa. Navigoi kohtaan ”Mouse & Keyboard” ja valitse ”If Key Down”, raahaa logiikka vasemmalla olevalle tyhjälle alustalle. Näet logiikan kuvauksen siitä mikä on sen funktio, kun siirrät osoittimen kuvakkeen päälle. Valitse ”If Key Down” valikon ”Key” kohtaan ylöspäin osoittava nuolinäppäin klikkaamalla pienempää kuvaketta tekstirudun vierestä ja valitse ”Up”. Seuraavaksi lisätään funktioon liikettä raahaamalla ”Movement” valikosta kohta ”Add Motion”. Varmista että siirrät ”Add Motionin” vastaavalla tavalla, eli ”If Key Down” kohdan oikealle puolelle, jolloin vihreä viiva osoittaa sivuun. Lisätään kuvaan ”Direction” eli suunta, johon liikettä lisätään, tässä kohtaa käytämme komentoa ”image\_angle” eli kuvan suuntaa ja lisäämme nopeutta arvolla 0.1.

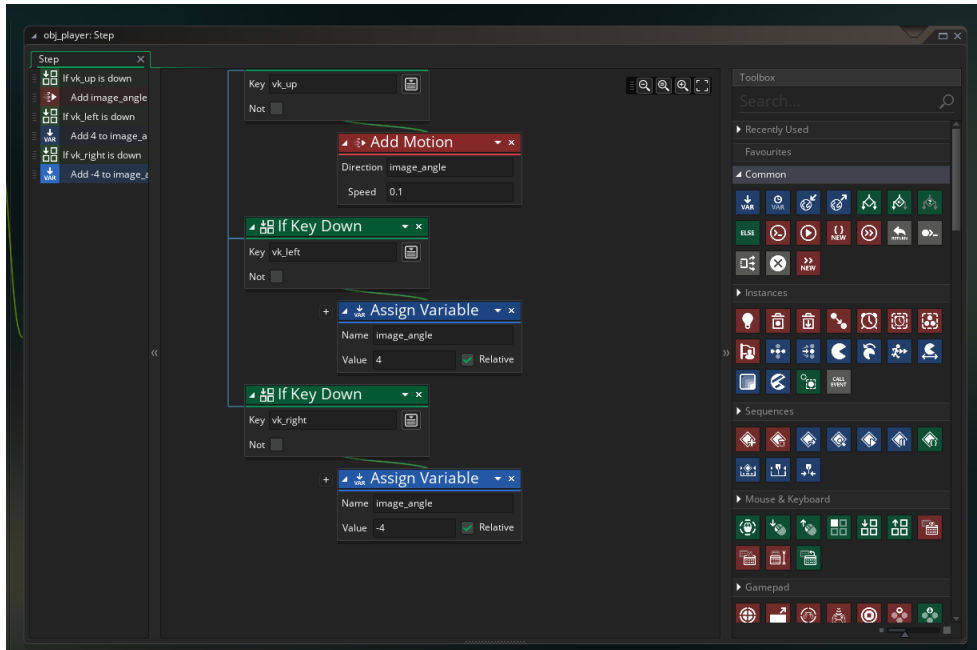


Kuva 13. Kuvakaappaus lohkojen lisäämisestä step-eventtiin

Seuraavaksi lisäämme objektille vielä kaksi logiikkaa kääntymistä varten. Kuva 13 havainnollistaa lohkojen lisäämistä step-eventtiin. Lisää samalla tavalla ”If Key Down” -funktio, mutta raahaa nämä ensimmäisen vihreällä värillä olevan ”If Key Down” -lohkon päälle, jolloin saat sinisen viivan, joka määrittelee sen, miten ohjelmisto tulkitsee logiikkaa. Vaihda toiseen ”If Key Down” -funktioon vasen nuolinäppäin ”vk\_left” ja toiseen ”vk\_right”.

Näiden lisäksi lisätään molempiin muuttuja ”Assign Variable” -logiikalla. Näiden ”Name” -kohtaan lisätään ”image\_angle”, eli kuvaa liikutetaan tietyn määrän verran tiettyyn suuntaan. Annamme toiselle arvoksi positiivisen luvun eli 4 ja toiselle negatiivisen eli -4. Muista täpätä molempiin kohtaan ”Relative”. Voit nyt klikata ylävalikosta kohtaa ”Play” tai painaa

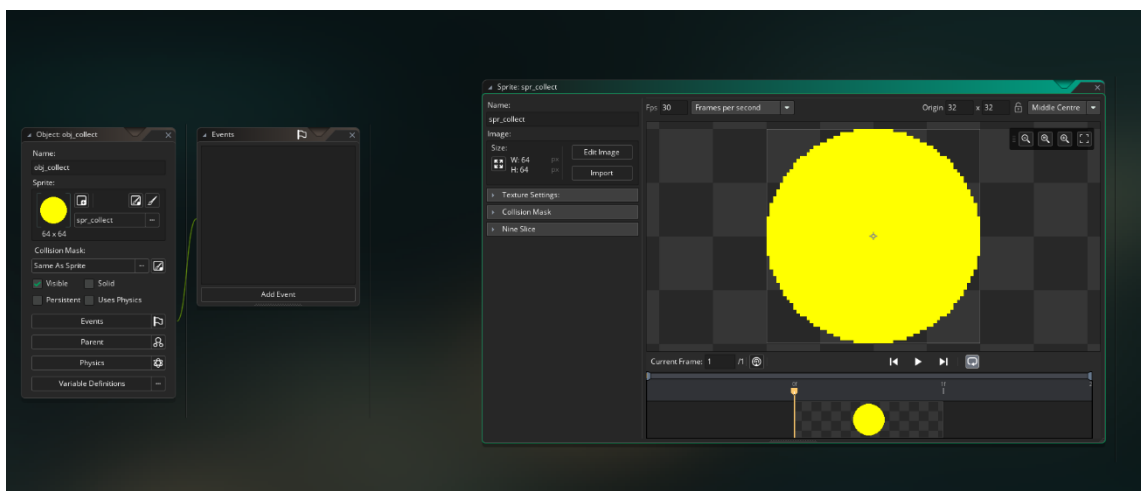
pikanäppäintä ”F5” ja testata peliä.



Kuva 14. Kuvakaappaus valmiista pelihahmon liikkumiseen liittyvästä logiikasta

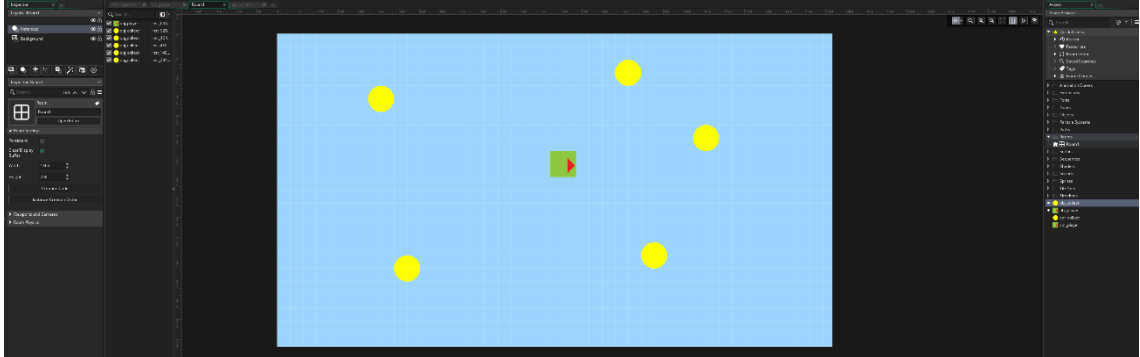
#### 4.3 Kerättävien kohteiden logiikka

Luodaan seuraavaksi hyvin yksinkertainen logiikka kerättävien objektien osalta. Tehdään samalla periaatteella uusi sprite sekä objekti, jotka voidaan nimetä esimerkiksi ”spr\_collect” ja ”obj\_collect”. Kuva 15 havainnollistaa valmiin kerättävän objektin kuvan luomista. Tehdään spritestä esimerkin vuoksi keltainen ympyrä käyttämällä ”Ellipse Toolia” (pikanäppäin ”C”). Lisätään myös kuvan ”Origin” pisteeksi ”Middle Center”.



Kuva 15. Kuvakaappaus valmiista kerättävän objektin kuvasta

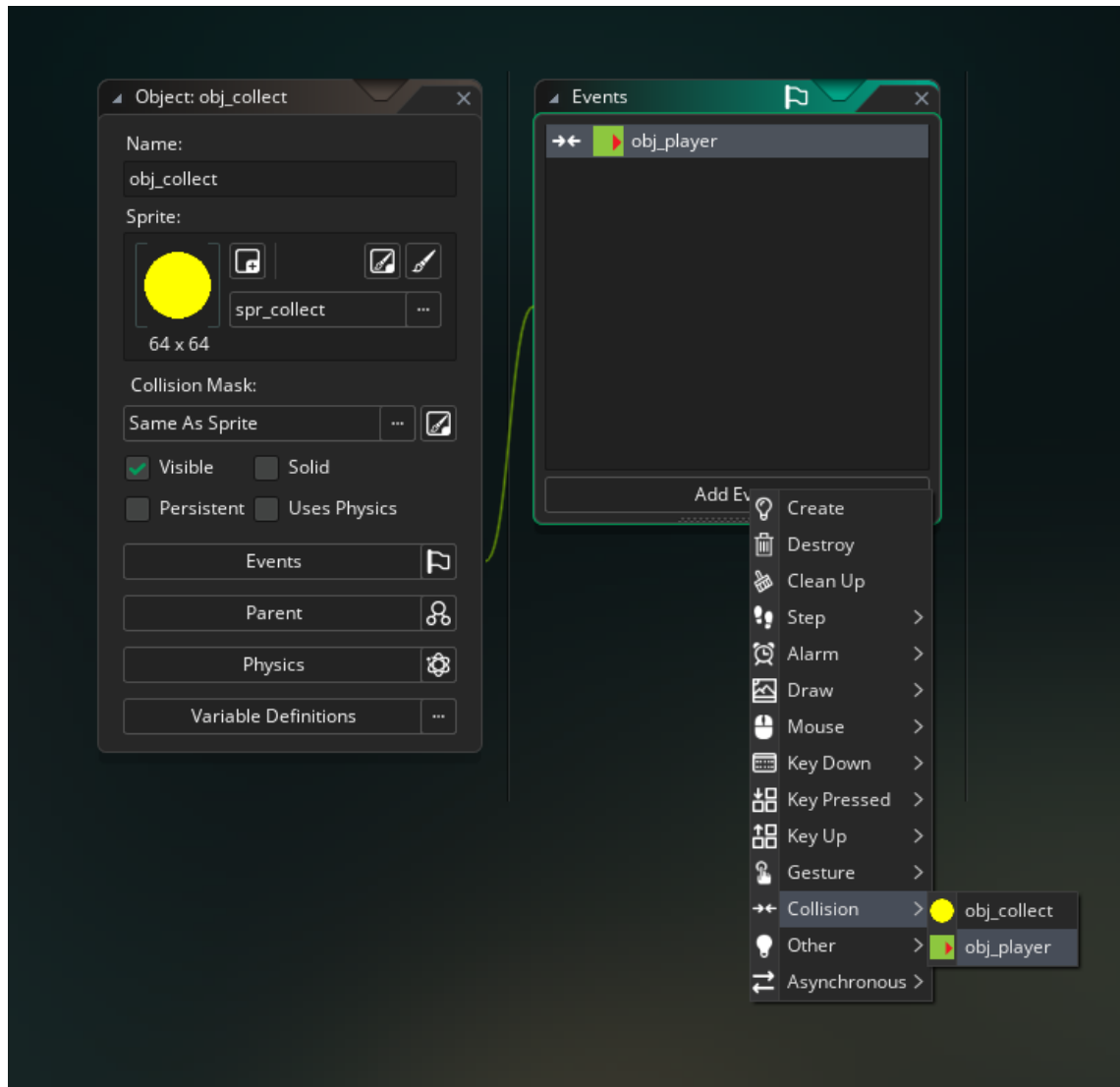
Siirrytään Room Editoriin ylävalikosta ja lisätään uudet “collect”-objektit peliruudulle raahaamalla ne oikealla olevasta valikosta. Kuva 16 havainnollistaa huonetta, johon collect-objektit on lisätty.



Kuva 16. Kuvakaappaus huoneesta johon on lisätty collect-objekteja

Palataan nyt tämän uuden objektin logiikan pariin. Lisätään “obj\_collect” seuraava funktio eli “Collision,” joka määrittelee, mitä tapahtuu, kun kyseinen objekti kohtaa toisen objektin. Kuva 17 havainnollistaa collision-tapahtuman lisäämistä obj\_collectin ja obj\_playerin välille.

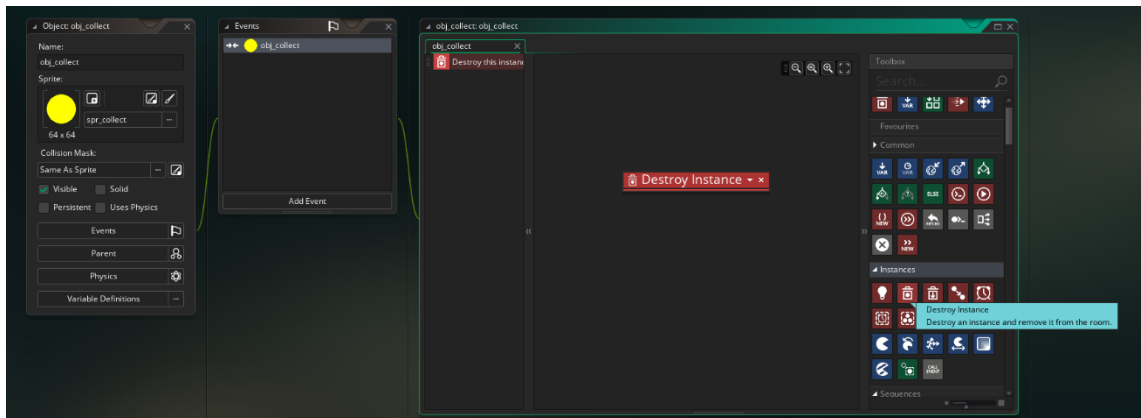
Valitaan toiseksi objektiksi pelihahmo, eli “obj\_player.”



Kuva 17. Kuvakaappaus kun lisätään collision-tapahtuma obj\_collectin ja obj\_playerin välille

Lisätään funktio “Destroy Instance” valikosta “Instances”. Kuva 18 havainnollistaa funktion “Destroy Instance” lisäämistä obj\_collectiin, jolloin objekti tuhoutuu, kun se törmää

pelihahmoon. Testataan peliä painamalla “Play” tai vaihtoehtoisesti pikanäppäintä “F5”.



Kuva 18. Kuvakaappaus funktion "Destroy Instance" lisäämisestä obj\_collecttiin

## 5 Konseptin testaaminen käytännössä

Konseptia testattiin kahdella ala-asteikäisellä, kahdeksan- sekä kaksitoista-vuotiailla, joilla on molemmilla taustaa videopelien pelaamisesta mutta ei pelien tekemisestä. Molemmat saivat käyttöönsä kannettavan tietokoneen, johon ohjelma oli jo etukäteen asennettu. Peliprojektia rakennettiin käytännössä opinnäytetyöntekijän ohjeistuksella perustuen kuitenkin tässä opinnäytetyössä olevaan peliprojektiin. Testitilanteessa opinnäytetyön tekijä ohjasi oppilaita askel askeleelta.

Testauksen aikana kahdeksanvuotias oppilas oli innostunut pelin tekemisestä ja tykkäsi erityisesti mahdollisuudesta piirtää omia kuviaan sprite editorissa. Erityisesti pelihahmon tekeminen oli kahdeksanvuotiaalle erityisen mukavaa. Hänen mukaansa visuaalinen ohjelmointi tuntui helpolta, kun opinnäytetyön tekijä ohjasi jokaisessa vaiheessa. Haasteita tuli lähinnä siinä, kun piti muistaa, missä järjestyksessä asioita tehtiin. Seuraavassa versiossa voisi olla pieni muistilista tärkeimmistä vaiheista.

Kaksitoistavuotiaalle Game Maker-ohjelmisto oli selkeämpi, sen käyttäminen tuntui luontevammalta ja perusasioiden omaksuminen oli nopeampaa. Hänellä oli kuitenkin vaikeuksia ymmärtää, miten ja miksi pelin logiikka toimii. Hänen mukaansa opinnäytetyön tekijän tuki auttoi paljon. Oppimateriaaliin voisi lisätä selityksiä siitä, miksi asiat tehdään tietyllä tavalla.

Molemmille peliprojektin tekijälle oli selvästi hyötyä siitä että joku oli ohjaamassa ja näyttämässä miten asioita tehdään. He eivät välttämättä olisi ymmärtäneet kaikkia työvaiheita ilman erillistä opastusta.

Yhteenvetona molemmat oppilaat olivat tyytyväisiä kokemukseen pelin tekemisestä ja kertoivat, että oppiminen oli hauskaa. Taulukko 1. kuvaa miten kaksi eri testaaajaa kokivat pelinkehityksen.

Ikäryhmä	Havainnot	Jatkokehitystarpeet
8-vuotias	Innostui sprite editorin käytöstä ja hahmojen piirtämisestä	Muistilista tärkeimmistä työvaiheista auttaisi muistamaan tehtävien järjestyksen
	Visuaalinen ohjelmointi tuntui helpolta opastuksen ansiosta	
	Kokemus oli hauska ja motivoiva.	
12-vuotias	Omaksui perusasiat nopeasti ja koki ohjelmiston käytön selkeäksi.	Oppimateriaaliin tarvitaan lisäyksiä, joissa selitetään tarkemmin, miksi tiettyjä asioita tehdään tietyllä tavalla.
	Vaikeuksia ymmärtää pelin logiikan toimintaa ilman lisäselityksiä.	

Taulukko 1. Havainnollistava taulukko ikäryhmittäin

## 6 Pohdinta

Jatkokehitysmahdollisuuksia konseptin luomiseen on runsaasti. Testauksen palaute osoitti selkeästi, että yksinkertainen ja havainnollistava materiaali auttaa nuorempia oppijoita, mutta sen ei aina voida olettaa riittävän täysin itsenäiseen työskentelyyn. Opettajan tai ohjaajan tuki osoittautui merkittäväksi tekijäksi oppimisen sujuvuudessa. Tämä herättää kysymyksen: onko opettajan tai muun tuen tarve välttämätöntä, vai voisiko oppimateriaalia kehittää edelleen niin, että oppijat pärjäisivät ilman ulkopuolista ohjausta? Konseptia voisi testata laajemmin eri ikäryhmissä ja eritasoisilla oppijoilla. Näin saataisiin parempi käsitys siitä, millaiset ratkaisut toimivat ja missä tarvitaan lisäkehitystä.

Yksi keskeinen havainto oli, että reaaliaikainen tuki ja ohjaus autoivat erityisesti tilanteissa, joissa oppijat kokivat epävarmuutta työvaiheiden järjestyksestä tai pelin logiikan ymmärtämisestä. Tällaisessa tilanteessa visuaalisen ja selkeästi etenevän materiaalin rooli korostuu. Tulevaisuudessa voitaisiin kehittää esimerkiksi vuorovaikutteisia oppimisalustoja, jotka opastavat oppijoita askel askeleelta ja tarjoavat tukea heti ongelmatilanteissa. Tekoälypohjaiset apurit tai chat-tyyppiset ohjaajat voisivat täydentää perinteistä oppimateriaalia, vähentäen opettajan roolin merkitystä.

Materiaalia voisi täydentää selkeillä visuaalisilla muistilistoilla ja etenemiskaavioilla. Tämä auttaa oppijoita ymmärtämään paremmin, missä järjestyksessä asioita tehdään ja miksi. Jos konseptia kehittäisi siihen suuntaan jossa opettaja tai ohjaaja ei ole läsnä, voisi myös valmiiksi kuvatut opetusvideot tukea oppimista.

On kuitenkin mahdollista, että tietynlainen tuki on aina tarpeen, erityisesti nuorempien oppijoiden kohdalla. Tämä ei kuitenkaan ole pelkästään heikkous vaan myös mahdollisuus, sillä henkilökohtainen ohjaus voi lisätä oppijoiden sitoutumista ja tehdä oppimiskokemuksesta mielekkäämmän. Tulevaisuudessa voisi pohtia, miten opettajien tai ohjaajien rooli voitaisiin optimoida esimerkiksi siten, että he voivat keskittyä monimutkaisempiin ongelmiin ja antaa oppijoiden suorittaa yksinkertaiset tehtävät itsenäisesti.

Tavoitteena oli luoda oppimismateriaali, jonka avulla oppijat voivat kokea onnistumisen iloa ja löytää omaa luovuuttaan pelinkehityksen kautta. Mielestäni opinnäytetyö onnistui tässä erittäin hyvin. Tulevaisuuden visio voisi sisältää esimerkiksi verkkoalustan, jossa opetusmateriaalit, vuorovaikutteiset työkalut ja yhteisön tuki yhdistyvät. Tämä tekisi pelinkehityksen oppimisesta saavutettavaa ja innostavaa yhä suuremmalle joukolle. Samalla voitaisiin edistää pelinkehitystä matalan kynnyksen harrastuksena, joka tukee lasten ja nuorten teknisiä, luovia ja ongelmanratkaisutaitoja.

## Lähteet

GameMaker. 2024. Make 2D Games With GameMaker. Saatavilla: <https://gamemaker.io/>. (Viitattu: 11.11.2024).

YoYo Games, 2024. GameMaker Manual. *Introduction To GameMaker*. Saatavilla osoitteessa: <https://manual.yoyogames.com> (Viitattu 11.11.2024).

Oulun yliopisto. 2024. Pelikehityksen hyödyntäminen ohjelmoinnin opetuksessa. Saatavilla: <https://oulurepo.oulu.fi/bitstream/handle/10024/48891/nbnfioulu-202404192863.pdf> (Viitattu 21.11.2024).

Jyväskylän yliopisto. 2024. Videopelit oppimisen tukena: opettajien ja peliavusteisen opetuksen asiantuntijoiden näkemyksiä. Saatavilla: <https://jyx.jyu.fi/bitstream/handle/123456789/87397/URN%3ANBN%3Afi%3Aju-202306023451.pdf> (Viitattu 21.11.2024).

Yleisradio. 2024. Pelit opettavat ongelmanratkaisua - tätä peliä kannattaa suositella lapsellesi, koska hän tarvitsee sen taitoja tulevaisuudessa. Saatavilla: <https://yle.fi/aihe/a/20-10006232> (Viitattu 21.11.2024).

## Kuvat

Kuva 1. Kuvakaappaus projektin luonnin näkymästä .....	10
Kuva 2. Kuvakaappaus projektin luonnista ja sen tyypistä .....	11
Kuva 3. Kuvakaappaus workspace näkymästä .....	11
Kuva 4. Kuvakaappaus ensimmäisen spriten luomisesta .....	12
Kuva 5. Kuvakaappaus sprite editorista .....	12
Kuva 6. Kuvakaappaus sprite editorilla tehdystä hahmosta .....	13
Kuva 7. Kuvakaappaus objektin luomisesta pelihahmolle .....	13
Kuva 8. Kuvakaappaus spriten lisäämisestä pelihahmon objektiin .....	14
Kuva 9. Kuvakaappaus huone-editorista ja taustaväriin muuttamisesta .....	14
Kuva 10. Kuvakaappaus jossa pelihahmon objekti on siirretty huoneeseen .....	15
Kuva 11. Kuvakaappaus ensimmäisen actionin lisäämisestä pelihahmon objektiin .....	15
Kuva 12. Kuvakaappaus Scripting preferensseistä .....	16
Kuva 13. Kuvakaappaus lohkojen lisäämisestä step-eventtiin .....	17
Kuva 14. Kuvakaappaus valmiista pelihahmon liikkumiseen liittyvästä logiikasta .....	18
Kuva 15. Kuvakaappaus valmiista kerättävän objektin kuvasta.....	18
Kuva 16. Kuvakaappaus huoneesta johon on lisätty collect-objekteja .....	19
Kuva 17. Kuvakaappaus kun lisätään collision-tapahtuma obj_collectin ja obj_playerin välille .....	20
Kuva 18. Kuvakaappaus funktion "Destroy Instance" lisäämisestä obj_collectiin.....	21

## Taulukot

Taulukko 1. Havainnollistava taulukko ikäryhmittäin .....	22
---	----