



Tasohyppelypelin kehittäminen Unityllä

Salla Komppa

Haaga-Helia ammattikorkeakoulu

Tradenomi

Amk-opinnäytetyö

2025

Tiivistelmä

Tekijä Salla Komppa
Tutkinto Tradenomi
Raportin/Opinnäytetyön nimi Tasohyppelypelin kehittäminen Unityllä
Sivu- ja liitesivumäärä 37 + 2
<p>Tämän opinnäytetyön innoittavana taustatekijänä oli opinnäytetyön tekijän kiinnostus pelejä ja pelialaa kohtaan. Opinnäytetyön tavoitteena oli kehittää julkaisukelpoinen peli Unity-pelimoottoria käyttäen. Työ rajattiin käsittelemään kehitettävän pelin tekemisen lisäksi kehitystyössä käytettäviä ohjelmoinnin ja versionhallinnan työvälineitä.</p> <p>Pelin kehitystyö tehtiin Unity-pelimoottoria, ja siihen suositeltua Visual Studio-tekstieditoria käyttäen. Visual Studiossa tehdyt ohjelmakoodit kirjoitettiin C#- eli C Sharp-ohjelmointikielellä. Työskentelyn aikana hyödynnettiin GitHub-pilvipalvelua ja GitHub Desktop-sovellusta pelitiedostojen versionhallintaan. Versionhallinnalla pystyttiin takaamaan pelitiedostojen säilyvyys ja mahdollisuus palata pelin aiempaan versioon. Työhön sisältyi myös pelisuunnittelua, peliä varten laadittiin pelikonsepti, joka sisältää tarkan kuvauksen pelin konseptista, sekä pelin käyttöliittymästä ja pelikehityksessä käytettävistä resursseista.</p> <p>Pelin kehittäminen aloitettiin pelihahmona toimivan shiba-rotuisen koiran luomisella. Hahmoon lisättiin animaatiot paikallaan seisomiselle, juoksemiselle, hyppäämiselle sekä putoamiselle. Hahmolle lisättiin myös kyky liikkua ja hyppiä pelikentässä. Hahmon jälkeen peliin luotiin ikuisesti jatkuva taustakuva, joka liikkuu pelaajan liikkeitä seuraten. Taustakuvan lisäksi pelikenttään luotiin 25 erilaista tasomallia, joiden päällä pelihahmon tulee hyppiä edetäkseen kentässä. Tasomallien kenttään ilmestymisen järjestys ohjelmoitiin satunnaiseksi. Jokaisen alustan yläpuolelle kehitettiin alustojen lailla satunnaisesti ilmestyviä, pelaajan kerättäviä timantteja ja väisteltäviä piikkiesteitä. Peliin tehtiin käyttöliittymä, joka sisältää pelin pysäytys- ja lopetusikkunat, päävalikon sekä pelaajalle näytettävän tiedon pelihahmon elämien ja kerättyjen timanttien määräästä. Pelin päävalikkoon lisättiin hahmokauppa, josta pelaaja voi vaihtaa pelihahmona olevan koiran väriä ostamalla väri vaihtoehtoja keräämiään timantteja vastaan. Lopuksi peliin lisättiin taustamusiikki ja äänitehosteita pelin painikkeille, sekä pelihahmon hyppäämistä ja esineisiin törmäyksiä varten.</p> <p>Pelin kehityksen aikana järjestettiin kaksi pelitestausta, joissa 8 hengen testiryhmälle jaettiin linkki pelin testiversioon, sekä Webropol-sivustolla luotuun testauskyselyyn. Ensimmäinen testaus keskittyi pelin kriittisten toimintojen, kuten hahmon liikuttamisen ja animaatioiden toimintaan, pelissä kerättävien timanttien ja väisteltävien esteiden sijaintiin, sekä pelihahmon kuoltua pelin uudelleenkäynnistämispainikkeen toimivuuteen. Toinen testaus keskittyi pelin taustamusiikin ja äänitehosteiden toimintaan, pelin valikkoihin sekä pelihahmon vaihtamiseen. Testauskyselyiden perusteella peliin tehtiin muutamia pieniä muutoksia, ennen kuin sen lopullinen versio ladattiin itch.io-sivustolle.</p>
Asiasanat Peliohjelmointi, ohjelmointikieli, Unity-pelimoottori, pelisuunnittelu, tasohyppelypeli

Sisällys

1	Johdanto.....	1
2	Peliohjelmointi	3
2.1	Unity	3
2.2	Visual Studio	4
2.3	C#.....	5
2.4	GitHub-versionhallinta	5
3	Pelisuunnittelu	7
4	Shiba & Sparkle-tasohyppelypeli	11
4.1	Shiba & Sparklen pelikonsepti.....	11
4.2	Vuokaavio	12
4.3	Pelin käyttöliittymä	14
4.4	Peliresurssit	14
5	Pelin toteutus.....	16
5.1	Hahmon luominen.....	16
5.2	Kentän luominen.....	17
5.3	Pelinäkymän käyttöliittymä	23
5.4	Valikko ja ponnausikkunat.....	24
5.5	Hahmokauppa.....	24
5.6	Äänet.....	25
6	Pelin testaus ja julkaisu.....	27
6.1	Ensimmäinen testaus	27
6.2	Toinen testaus	28
6.3	Testaustulosten perusteella tehdyt muutokset peliin	30
6.4	Pelin julkaisu	30
7	Pohdinta	31
	Lähteet	34
	Liitteet.....	38
	Liite 1. Ensimmäisen pelitestauksen testauskysymykset.....	38
	Liite 2. Toisen pelitestauksen testauskysymykset.....	39

1 Johdanto

Tämän opinnäytetyön aiheena on tasohyppelypelin kehittäminen Unity-pelimoottorilla. Valitsin pelin kehittämisen aiheekseni, sillä olen kiinnostunut pelien tekemisestä, ja suunnittelen työllistyväni pelialalle opintojeni päätyttyä. Minulla ei ole aiempaa kokemusta pelin julkaisemisesta, joten olen kiinnostunut oppimaan kattavammin pelin kehittämisen eri vaiheista, suunnittelusta julkaisuun. Opinnäytetyöllä ei ole toimeksiantajaa. Opinnäytetyön tuotoksena tehtävä peli on suurin peliprojekti, mitä olen tähän asti itsenäisesti tehnyt, ja saankin siitä hyvän pohjan henkilökohtaiselle peliportfoliolleni. Suomen peliala on kokenut suurta kasvua viime vuosien aikana, joten aiheeni on ajankohtainen. Tästä työstä hyötyvät peleistä ja pelialasta kiinnostuneet henkilöt, jotka ovat kiinnostuneita pelin suunnittelusta, kehittämisestä ja julkaisusta.

Työ on rajattu käsittelemään tuotoksena olevan pelin tekemisessä käytettäviä työvälineitä, sekä yksin verkkoselaimella pelattavan pelin kehitystä. Työtä ei ole tarkoitettu yksityiskohtaiseksi pelikehitysohjeeksi, vaan siinä kerrotaan päällisin puolin pelikehityksen eri vaiheista. Työn ulkopuolelle on rajattu moninpelit, kaupalliset pelit, sekä erilaisille konsoleille tehtävät pelit. Opinnäytetyön aikana ei myöskään tehdä itse pelissä käytettäviä materiaaleja yhtä ääniefektiä lukuun ottamatta, vaan grafiikoihin ja ääniefekteihin käytetään internetistä valmiiksi löytyviä materiaaleja. Materiaalien käytössä toimitaan vastuullisesti tekijänoikeus huomioon ottaen: jokaisen materiaalin käyttöoikeudet tarkistetaan ennen materiaalin käyttöönottoa. Peliin ei myöskään sisällytetä laittomia aiheisältöjä.

Opinnäytetyön tavoitteena on kehittää julkaisukelpoinen tasohyppelypeli. Työhön liittyviä tutkimuskysymyksiä ovat; minkälaisia ohjelmistoja pelin kehittämisessä tarvitaan, mitä pelisuunnittelussa tulee ottaa huomioon, kuinka resurssien rajallisuus vaikuttaa pelin kehittämiseen, sekä miten pelin testauttaminen vaikuttaa pelin kehitykseen?

Pelin kehitystyö jaetaan osiin: ensin tehdään pelin suunnittelu, jonka jälkeen aloitetaan pelin ohjelmointi. Kun pelistä on prototyyppi valmiina, se testautetaan testiryhmällä. Testitulosten perusteella peliin tehdään tarvittavia muutoksia, ennen kuin pelin kehitystä jatketaan. Ennen pelin julkaisua järjestetään vielä yksi testaus, jossa ilmenevät ongelmat pyritään korjaamaan ennen pelin lopullista julkaisua. Pelin julkaisemisen jälkeen pohdin omaa oppimistani pelikehitysprosessin aikana.

Työhön liittyy erilaisia pelikehityksen keskeisiä käsitteitä. **Peliohjelmoinnilla** tarkoitetaan pelien luomista, johon voi sisältyä ohjelmoinnin lisäksi myös esimerkiksi graafista suunnittelua. Peliohjelmoinnissa käytetään siihen erikseen kehitettyjä ohjelmia ja ohjelmointikieliä. (Trang 3.7.2024.)

Ohjelmointikielen avulla ohjelmoijat pystyvät antamaan tietokoneille suoritettavia tehtäviä. Kieliä on useita erilaisia, joista kaikilla on oma rakenne ja sääntönsä. (Trang 7.7.2024.) Tämän opinnäytetyön tuotoksena kehitettävä peli tehdään peliohjelmointia varten kehitetyllä **Unity-pelimoottorilla**, joka on yksi maailman johtavista pelimoottoreista (Lanzinger 2021, luku 1.1).

Pelisuunnittelun avulla kehitteillä olevalle pelille voidaan laatia kehys, joka sisältää esimerkiksi pelin säännöt, pelimekaniikat ja tavoitteet. (Kramarzewski & De Nucci 2023, luku 1.) **Tasohyppelypelissä** pelaajan tavoitteena on kulkea pelihahmolla erilaisten kenttien läpi, hyödyntäen hahmon liikkeitä. Pelikentissä on usein tasoja, joille pelihahmon täytyy hypätä. (MasterClass 2021.)

2 Peliohjelmointi

Peliohjelmointi on pelien kehitystyötä erillisillä, pelien tekoa varten suunnitelluilla ohjelmistoilla. Peliohjelmointityö sisältää useita erilaisia osa-alueita, kuten suunnittelutyötä, ohjelmointia, pelin toiminnan testaamista sekä erilaisten päivityksien ja parannuksien tekoa jo julkaistuun peliin. Pelisuunnittelijoiden käytettävissä on lukuisia erilaisia ohjelmointikieliä, pelimoottoreita ja työskentelyä tukevia sovelluksia, kuten tekstieditoreita koodin kirjoittamiseen, grafiikkatyökaluja kuvankäsittelyä ja 3D-mallien luomiseen sekä versionhallinnan työkaluja. (Trang 3.7.2024.) Tässä raportissa keskitytään ainoastaan tämän opinnäytetyön toiminnallisen osion, Shiba & Sparkle-pelin kehityksessä käytettävään Unity-pelimoottoriin, Visual Studio-tekstieditoriin, C#-ohjelmointikieleen sekä GitHub-versionhallintaan.

2.1 Unity

Kööpenhaminalaisten Joachim Anten, Nicholas Francisin sekä David Helgasonin perustama Unity Technologies kehitti Unity-pelimoottorin, jonka ensimmäinen versio julkaistiin kesäkuussa 2005. Unity oli alun perin suunniteltu Mac-alustoilla käytettäväksi, ja sillä pystyikin aluksi kehittämään pelejä ainoastaan Mac OS X-käyttöjärjestelmällä pelattavaksi. Julkaisusta oli kulunut aikaa vuosi, kun Unity sai päivityksen, joka mahdollisti pelien kehittämisen Macin lisäksi myös Windowsilla ja erilaisilla verkkoalustoilla pelattavaksi. Vuonna 2008 Applen julkaisema App Store-sovelluskauppa mahdollisti mobiilipelien nousun pelaajien suosioon, johon Unityn kehittäjät reagoivat nopeasti mahdollistaen Unityllä kehitettävien pelien pelaamisen Applen iOS-käyttöjärjestelmällä. Muutamaa vuotta myöhemmin myös Android-käyttöjärjestelmällä pelattavia pelejä oli mahdollista kehittää Unityllä. (Cohen-Peckham 2019.)

Nykypäivänä Unity on yksi maailman johtavista pelimoottoreista. Kännyköiden ja tietokoneiden käyttöliittymien lisäksi Unityn pelikehitysskaala on laajentunut myös pelikonsolien ja virtuaaliodellisuuden pariin. (Lanzinger 2021, luku 1.1.)

Unity tarjoaa käyttäjilleen erilaisia tilausvaihtoehtoja riippuen heidän käyttötarpeistaan. Ilmainen Unity Personal-tilausmalli soveltuu kaikille, jotka haluavat tehdä pelejä kokeilu- tai harrastusmielessä, tai ovat osana pientä pelikehitysorganisaatiota. Mikäli yksilön tai organisaation tuotto Unityllä kehitetyistä peleistä ylittää 100 000 dollarin rajan, tulee tilausmalli muuttua maksulliseen yritysversioon. (Hardman 2024, luku 1.1.) Maksullisiin tilausvaihtoehtoihin sisältyy Unityn tekninen tuki. Opiskelijoille on vaihtoehtona Unity Personal-tilausmallin lisäksi myös ilmainen Unity Student-tilausmalli, johon sisältyy 20 % alennus Unity Asset Storesta ostettaviin peliresursseihin. Saatavilla on myös muutama tilausmalli Unity-koulutautumiseen liittyen. (Unity Technologies s.a. a..)

Unityn oma Asset Store julkaistiin vuonna 2010. Kyseessä on Unityn käyttäjäkunnalle kehitetty kaupankäyntialusta, joka mahdollistaa pelikehityksessä käytettävien peliresurssien, ohjeiden ja kirjastojen ostamisen, myymisen sekä ilmaiseksi jakamisen ja lataamisen. (Unity Technologies 2010.) Vuonna 2020 julkaistu Unity-päivitys muutti Asset Storen käytettävyyttä siirtämällä sen Unityn sisäisestä ikkunasta erilliseen, pelimoottorin ulkopuoliseen verkkosivustoon (Unity Technologies s.a. b). Unity Asset Storessa jaossa olevien peliresurssien käyttöturvallisuus on tarkasti Unityn valvonnan alaisena (Unity Technologies s.a. c).

Unityn käyttöönoton yhteydessä käyttäjä luo oman Unity-tunnuksensa, johon käyttäjän valitsema Unity-tilausmalli yhdistyy. Tunnuksen luomisen jälkeen tulee asentaa Unity Hub-sovellus, jonka kautta Unityn saa asennettua. Unity Hubin kautta käyttäjä pystyy myös asentamaan Unityyn julkaistavat päivitykset ja uudet versiot, sekä tarvittaessa ottamaan käyttöön vanhempia Unity-versioita. Jokaisen version nimeen on selkeästi merkitty version julkaisuvuosi. Joissain versionimissä on myös termi LTS, joka on lyhenne sanoista Long Term Support, eli pitkän aikavälin tuki. LTS-versio on turvallinen vaihtoehto etenkin julkaistavaksi tarkoitetuille peleille, sillä kyseiseen Unity-versioon ei tule myöhemmin päivityksiä, jotka voisivat vaarantaa pelin toimintakykyä tai aiheuttaa tarvetta suurien muutosten tekemiseen. (Hardman 2024, luku 1.1.)

Asennuksen yhteydessä Unityn toiminnan tueksi voi asentaa erilaisia moduuleja, jotka tukevat pelikehitystä eri käyttöjärjestelmille, kuten pelikonsoleille tai mobiililaitteille. Tärkein lisättävä moduuli on tekstieditori, jossa peliin tarvittavat koodit kirjoitetaan. Joillakin tekstieditorilla pystyy myös tekemään vianmääritystä peliin. Unity ehdottaa asennuksensa yhteydessä ensisijaiseksi tekstieditoriksi Microsoftin kehittämää Visual Studiota. (Hardman 2024, luku 1.1.)

2.2 Visual Studio

Visual Studio on Microsoftin kehittämä tekstieditori, jonka avulla kehittäjät voivat tehdä pelien lisäksi graafisia käyttöliittymiä ja sovelluksia erilaisille käyttöjärjestelmille. Visual Studio tukee useita erilaisia ohjelmointikieliä. Ensimmäinen Visual Studio-versio julkaistiin vuonna 1997. Ilmainen versio, Visual Studio Community, julkaistiin vuonna 2014. (GeeksForGeeks 2023.)

Visual Studiosta on tarjolla kolme erilaista tilausmallia, joista kaksi, Visual Studio Enterprise sekä Visual Studio Professional, on kehitetty yritystoimintaa varten. Yrityslisenssin käyttöönottovaatimuksen määrittelyyn on kaksi erilaista yritystoiminnan rajausta: Visual Studiota käyttävien työntekijöiden määrä, sekä yrityksen vuositulot. Mikäli yli 250 työntekijää käyttää Visual Studiota, tai yrityksen vuositulot ylittävät miljoona dollaria, yrityksen on käytettävä maksullista Visual Studio-versiota. Suurille yrityksille tarkoitettu Visual Studio Enterprise on kaikista Visual Studio-versioista kattavin. Visual Studio Professional soveltuu pienempien yritysten käyttöön, ja se sisältää saman rajoitetun

sisällön, kuin ilmaiseksi ladattava Visual Studio Community. Visual Studio Communityn lisenssiä voi käyttää maksimissaan 5 käyttäjää samasta organisaatiosta. Kaikista Visual Studio-versioista kattavin on Visual Studio Enterprise, joka on tarkoitettu suurien yritysten käyttöön. Pienemmille yrityksille tarkoitettu Visual Studio Professional sisältää samat rajatut ominaisuudet, kuin ilmainen Visual Studio Community. (Microsoft s.a..)

2.3 C#

C#, lausutaan C Sharp, on Anders Hjelbergin Microsoftille kehittämä ohjelmointikieli, joka on Unity-ohjelmoinnissa eniten käytävä ohjelmointikieli (Designveloper 15.10.2024; Evans 2.6.2023.). Alun perin Cool-nimellä tunnettu C# julkaistiin vuoden 2000 heinäkuussa. Peliohjelmoinnin lisäksi C# on laajassa käytössä esimerkiksi erilaisten ohjelmistojen ja verkkosivujen luonnissa. (ACG Technical Editors Team 8.6.2023.)

C# on tarkoitettu objektiorientoituneeksi ohjelmointitavaksi (Designveloper 15.10.2024). Objektilla tarkoitetaan kaikkia osia, joita pelin näkymässä voi olla, kuten pelihahmoja tai käytettäviä esineitä. Jokaisen objektin toimintaan ja ulkonäköön vaikuttavat komponentit ovat sidoksissa objektiin. (Unity Technologies s.a. d..) Objektien avulla kehitettävän pelin saa jaettua pienempiin kokonaisuuksiin. Pienemmät kokonaisuudet mahdollistavat helpomman tavan pelikokonaisuuden hallintaan, kokoamiseen ja testaamiseen, sekä tekevät pelin toiminnan testaamisesta selkeämpää. (Designveloper 15.10.2024.)

2.4 GitHub-versionhallinta

GitHub on vuonna 2008 perustettu versionhallintaa tarjoava verkkopalvelu. GitHubin toiminta rakentuu suomalaisen Linus Torvaldsin vuonna 2005 kehittämään Git-versionhallintapalvelun ympärille. Palveluiden tärkeimpinä eroina on sijainti sekä turvallisuus. Git on tietokoneelle paikallisesti asennettava, useimmiten tietokoneen komentokehoteen kautta käytettävä ohjelma, GitHub puolestaan on verkon välityksellä toimiva pilvipalvelu. Turvallisuudessa Git ja GitHub eroavat tunnistautumisessa: Gitissä jokainen, joka pääsee tietovarastoon käsiksi voi muokata ja poistaa mitä tahansa tietovaraston osia vapaasti. GitHub vaatii tunnistautumisen ja tietovaraston käyttöoikeuden, jotta muokkauksia ja poistoja voidaan suorittaa. (McKenzie 2024.)

Versionhallinta on niin sanottu turvaverkko ohjelmistokehityksen tueksi, jonka avulla ohjelmistokehittäjä pystyy palauttamaan aiempia versioita tekemästään koodista, esimerkiksi kriittisten virheiden tapahtuessa. Aiempien versioiden tallentamisen lisäksi versionhallinta mahdollistaa myös uusien toimintojen luomisen ja testaamisen ilman muille koodin osille aiheutuvaa riskiä. Tämän mahdollistaa versionhallinnan haarat. (GitLab Inc. s.a..)

Kun projektin versionhallinta aloitetaan, sille muodostuu automaattisesti päähaara, josta voidaan luoda uusia haaroja. Uusi haara sisältää aina kaiken koodin, mitä sen lähdehaara on luomishetkellä sisältänyt. Uuteen haaraan tehdyt muutokset eivät kuitenkaan toimi tai näy alkuperäisessä haarassa ennen kuin ne on yhdistetty toisiinsa. (Radigan s.a..)

Haaroissa työskentely mahdollistaa turvallisen toimintavan lisäksi myös nopeamman kehitystyön, sillä useat kehittäjät voivat samanaikaisesti työskennellä eri osa-alueiden parissa ilman ylimääräistä odottelua. Virhetilanteissa edellisen version uudelleen käyttöönottoaminen on nopeaa, eikä viallisen koodin selvittelytyö hidasta muiden tiiminjäsenten toimintaa. (GitLab Inc. s.a..) Saman tiimin jäsenet pystyvät myös työskentelemään toistensa luomissa haaroissa saumattomasti (Radigan s.a.).

GitHub tarjoaa kehittäjille lukuisia versionhallinnan työkaluja (McKenzie 2024.). Niistä tämän raportin kannalta keskeisin on GitHub Desktop, joka helpottaa ja nopeuttaa kehitystyötä mahdollistamalla versionhallinnan suorittamisen painikkeita erillisessä käyttöliittymässä painamalla, perinteisen tekstisyötteisen komentokehoteen sijaan.

3 Pelisuunnittelu

Pelisuunnittelua voi organisaation koon tai tekeillä olevan projektin mukaan tehdä yksi tai useampi henkilö. Suuremmissa tiimeissä yksittäiset suunnittelun osa-alueet voivat jakautua suunnittelutiimin yksittäisten jäsenten vastuulle, pienemmissä tiimeissä yksi jäsen voi tehdä useamman, tai jokaisen, alueen suunnittelutyön. Pelisuunnittelijan tehtäviin voi kuulua muun muassa tekstimuotoisen pelisuunnitelman ja vuokaavion laatiminen, pelin kenttien, mahdollisten dialogien ja peliin myöhemmin lisättävien sisältöjen sekä käyttöliittymän suunnittelu. (Kramarzewski & De Nucci 2023, luku 1.) Käyttöliittymä on parhaimmillaan pelin ja pelaajan välisen kommunikaation tärkein väline, jonka suunnittelussa pitää olla tarkkana, ettei peli syötä pelaajalle liikaa informaatiota kerralla (Kemppainen 2019, 77). Yksi suunnittelijan tärkeimmistä tehtävistä on kommunikoida muiden kehitystiimin jäsenten kanssa peliin liittyvistä suunnitelmista (Kramarzewski & De Nucci 2023, luku 1).

Suunnittelija joutuu huomioimaan työssään myös muiden tekemiä suunnitelmia ja vaatimuksia. Omiin ideoihin ei ole kannattavaa kiintyä liikaa, sillä suuri osa ideoista ei välttämättä päädy osaksi lopullista projektia. (Kemppainen 2019, 40–41.) Suunnittelija saattaa myös itse hylätä omia ideoitaan ilman, että ne edes pääsevät osaksi suunnitteludokumentaatiota (Kemppainen 2019, 56). Myös pelin kohderyhmän halujen ymmärtäminen on tärkeä taito pelisuunnittelun kannalta, sillä peli ei menesty ilman siitä kiinnostuneita pelaajia. Pelisuunnittelija osittain myös määrittelee pelin kohderyhmää suunnitelmaa tehdessään. (Kemppainen 2019, 75; Kemppainen 2019, 101.)

Pelikehitykseen käytettävä aika on huomioitava pelisuunnittelussa ja kehitystyön aikataulutuksessa. Hyvin laadittu aikataulu helpottaa kehitettävien osa-alueiden lajittelua tärkeysjärjestyksen mukaisesti. Pelin tärkeimmät osa-alueet kehitetään ensin, jotta voidaan ehkäistä pelin toimintaa uhkaavia kriittisiä virheitä, mikäli pelin kokonaiskehitysaika loppuu kesken. (Kemppainen 2019, 98.) Joissain tapauksissa suunnittelijalta vaaditaan myös vaihtoehtoisen suunnitelman laatimista, mikäli jokin peliin suunniteltu asia osoittautuu mahdottomaksi toteuttaa (Kemppainen 2019, 122). Esimerkiksi ajan tai budjetin rajallisuus tai tekniset puutteet voivat hankaloittaa alkuperäisen suunnitelman noudattamista.

Pelikonsepti tarkoittaa pelisuunnittelijan pelistä kokoamaa kuvausta, joka sisältää tietoa esimerkiksi pelin ulkonäöstä, toiminnallisuuksista ja kohderyhmästä. Tekstimuotoinen pelikonsepti on yleensä 2–5 sivun pituinen. Konseptitekstin alkuun kirjoitetaan lukijan mielenkiintoa herättelevä peliesittely. Tämän lisäksi pelistä kirjoitetaan tarkempi kuvaus, jossa kerrotaan pelin tärkeimmät piirteet. Konseptiin tulisi sisällyttää myös selvennys pelin pääasiallisesta kohderyhmästä, alustasta tai alustoista, joille peli kehitetään, sekä liiketoimintamalli. Konsepti voidaan esittää myös dia- tai vaikkapa videomuodossa. Osa konseptin osa-alueista voi tulla muilta tiimin jäseniltä tai

yhteistyökumppaneilta. Pelisuunnittelija kokoaa konseptin kokonaisuudeksi, ottaen huomioon konseptin yleisön. (Kramarzewski & De Nucci 2023, luku 2.)

Peliesittely on tärkein pelikonseptin lukijan mielenkiinnon herättäjä. Jos esittely on tekstimuodossa, se on oltava lyhyt, maksimissaan kolmen lauseen pituinen. Esittelytekstiin sisällytetään kaikki pelikonseptin osa-alueet tiiviissä muodossa. Esittelytekstin voi korvata myös muilla esitysvaihtoehdoilla, kuten konseptitaiteella tai miellekartalla. (Kramarzewski & De Nucci 2023, luku 2.)

Konseptiin sisällytetään tarkka kuvaus pelin sisällöstä ja toiminnasta. Pelin juonen, sekä pelin tärkeimpien hahmojen ja pelimaailman tarinoiden kertominen auttaa lukijaa tai kuulijaa paremmin ymmärtämään konseptia. Lisäksi pelimuoto on tärkeää avata konseptissa. Tarkemman kuvauksen osiossa voi hyödyntää suurempaa määrää konseptitaidetta, kuin esittelyosiossa. Mikäli suunnitteluvaiheessa on jo tiedossa, minkälaista äänimaailmaa pelissä halutaan hyödyntää, sen voi myös esitellä tarkemmin. (Kramarzewski & De Nucci 2023, luku 2.)

Pelin tärkeimpiä ominaisuuksia kertoessa ei kannata mennä liian yksityiskohtaiseen kuvailuun. Tärkeimpiä kuvailtavia ominaisuuksia ovat pelaajan käytössä olevat hahmot, tekniset ominaisuudet, pelin etenemistaktiikka, sekä pelissä mahdollisesti oleva ainutlaatuinen ominaisuus, joka erottaa sen kilpailijoista. Kuvailtavia ominaisuuksia voi yhdistellä mielenkiintoa ylläpitäviksi iskulauseiksi. (Kramarzewski & De Nucci 2023, luku 2.)

Pelin suunnitteluun eniten vaikuttava asia, **pelialusta**, tarkoittaa laitteistoa, jolla peliä pelataan. Pelialustoja ovat esimerkiksi pelikonsolit, kuten PlayStation ja Xbox, tietokoneet sekä mobiililaitteet. (Kramarzewski & De Nucci 2023, luku 2.)

Pelialustojen välillä on erittäin paljon eroavaisuuksia esimerkiksi rajoitusten, ohjaustapojen ja liikuteltavuuden kannalta. Osa konsoleista on sidottu tiettyyn paikkaan, kuten esimerkiksi pelihallien peliautomaatit, tai televisioon kytkettävät pelikonsolit, ja osa pelialustoista voi kulkea mukana kaikkialle, kuten älypuhelimet ja Nintendo Switch. (Catness Game Studios 12.12.2024.)

Oikean pelialustan valitseminen pelille vaikuttaa melkein jokaiseen suunnitteluvaiheeseen, joista kenties tärkeimpänä pelin ohjattavuuteen ja suoritusvaatimuksiin. Pelialusta vaikuttaa huomattavasti myös kohderyhmän skaalaamiseen, sillä suunnitteilla oleva peli ei välttämättä olekaan toimiva ratkaisu esimerkiksi älypuhelimella, joten peliä ei kannata suunnitella mobiilipelaajia ajatellen. (Kramarzewski & De Nucci 2023, luku 2.)

Pelit voidaan luokitella useisiin erilaisiin **pelityyleihin**, joilla voi olla useita alakategorioita. Pelityylit ja -kategoriat antavat pelaajalle ennakkokäsityksen pelin sisällöstä ja mahdollisuuksista.

Esimerkiksi siitä, onko peli yksin- tai moninpeli voi osittain myös päätellä, tarvitseeko pelin pelaamiseen verkkoyhteyttä, eli onko kyseessä online- vai offline-peli. (Catness Game Studios 12.12.2024.)

Peliala kehittyä jatkuvasti, ja pelikehittäjät voivat vapaammin yhdistellä erilaisia lajityyppejä peleihinsä (Pavlovic 2020). Jotkin pelityylit ovat kuitenkin niin sanottuja vakioita, kuten esimerkiksi ta-sohyppelypeli, ammuntopeli tai simulaatio (Kramarzewski & De Nucci 2023, luku 2).

Pelin **kohderyhmä** koostuu sellaisista yksilöistä, jotka ovat potentiaalisesti kiinnostuneista pelistä tai ovat pelanneet samankaltaisia pelejä aiemmin. Kohderyhmän jäsenet ovat myös kiinnostuneita kuluttamaan peliin omaa aikaansa ja rahaansa, ja ovat usein uskollisia peliä kohtaan. (Ansbacher 25.3.2025; Kramarzewski & De Nucci 2023, luku 2.)

Tärkeä osa kohderyhmän rajaamisessa on myös pelaajien ikä. Etenkin alaikäisille tarkoitettujen pelien sisällön tulee sopia asetettuun ikärajaan, mikä rajoittaa suunnittelumahdollisuuksia (Kramarzewski & De Nucci 2023, luku 2).

Oikeanlaisen kohderyhmän rajaaminen auttaa menestyksekkään markkinointistrategian suunnittelussa ja toteutuksessa, vähentäen ylimääräisten kulujen muodostumista (Ansbacher 25.3.2025). Kohderyhmää rajatessa voidaan esimerkiksi pohtia pelaajien tarpeita, tehdä tutkimusta isoimmista kilpailijoista, tai palkata markkinoinnin ammattilaisia, joilla on tietoa erilaisista peleistä kiinnostuneista kohderyhmistä. (Ansbacher 25.3.2025; Xsolla 8.12.2023.)

Liiketoimintamallin avulla määritellään millä tavoin peli tuottaa rahaa. Peleihin liittyy kolme pääasiallista liiketoimintamallia: ennakkoon, kertaluontoisesti maksettava Premium-peli, kuukausimaksullinen peli sekä pienempiä mikromaksuja sisältävä ilmaispeli. Kertamaksullisten Premium-pelien liiketoimintamalli voi sisältää pelin hetkellistä ilmaisjakelua, joko koko pelin laajuudelta, tai pienempänä demomallina. Peliin voidaan myös julkaista lisäosia, joiden kautta pelaajat saavat lisää pelisisältöä, joko pientä maksua vastaan, tai täysin ilmaiseksi. (Kramarzewski & De Nucci 2023, luku 2).

Ilmaispelien tuotto voi koostua pelaajien suorittamista mikromaksuista tai pelissä näytettävistä mainoksista. Mainoksien kautta tuottoa voi kertyä esimerkiksi myymällä pelaajille mainosten poistamista, tai lukitsemalla jonkin pelin palkinnosta mainoksen katsomisen taakse. (Kramarzewski & De Nucci 2023, luku 2). Pelin jakelu ilmaispelina voi tuoda peliin todella suuria pelaajamääriä. Iso osa pelaajista ei ikinä kuluta pelissä omaa rahaansa, kun puolestaan osa pelaajista voi kuluttaa peliin suuriakin määriä rahaa. (Kramarzewski & De Nucci 2023, luku 15.)

Kuukausimaksuluonteiset pelit ovat enimmäkseen osana erilaisia pelipalveluita, joissa kuukausimaksua vastaan voi pelata palveluun sisältyviä pelejä. Kuukausimaksullinen pelipalvelu on

erinomainen vaihtoehto pelaajalle, joka haluaa pelata useita eri pelejä, sillä kuukausimaksupalvelu tulee edullisemmaksi, kuin ostaa jokainen peli erikseen. Kuukausimaksullisia pelipalveluita tarjoaa muun muassa Sony, Netflix ja Electronic Arts. (Minor 2025)

4 Shiba & Sparkle-tasohyppelypeli

Tämän opinnäytetyön toiminnallisena osana luodaan Unityllä ohjelmitava Shiba & Sparkle-tasohyppelypeli. Pelin inspiraationa toimii oma shiba-rotuinen koirani, Sipsi.

4.1 Shiba & Sparklen pelikonsepti

Mikä on pieni ja loistava – no tietenkin timantteja innokkaasti keräilevä shiba! Tule mukaan ilmaiseen, koko perheelle soveltuvaan tasohyppelyseikkailuun, jossa timantit ovat ikuisia. Verkkoeläimessä pelaaminen ei ole koskaan ollut näin suloista.

Shiba & Sparklen tavoite on yksinkertainen: pysy hengissä mahdollisimman pitkään, ja kerää niin monta timanttia kuin pystyt. Pelin päähahmona on suloinen, punainen shiba-rotuinen koira, jota pelaaja ohjaa loputtomasti jatkuvan tasohyppelykentän tasoja pitkin timanttien luo. Pelikenttä koostuu hypittävästä tasoista, palkinnoiksi kerättävistä timanteista, piikkiesteistä, liikkuvasta taustakuvasta, sekä ääniefekteistä. Ilmassa leijuvat tasohyppelytasot, joiden päälle pelihahmo hyppää, muodostuvat satunnaisesti, eri kokoisina ja eri korkeuksille. Alustojen väliin tulee satunnaisia aukkoja, joiden yli pelihahmon täytyy päästä seuraavalle alustalle. Jos pelihahmo putoaa aukkoon, peli loppuu välittömästi.

Palkinnoksi kerättävät timantit muodostuvat hypättävien tasojen läheisyyteen, jokaisesta timantista saa 1 pisteen. Jotta keräily ja kentässä eteneminen ei kävisi liian helposti, kenttään muodostuu satunnaisesti myös piikkiesteitä, jotka ovat pelihahmolle vaarallisia. Jos esteeseen törmätään, pelaaja menettää yhden kolmesta elämästään. Jos kaikki elämät on menetetty, peli loppuu. Liikkuva taustakuva koostuu kolmesta erillisestä kuvasta, jotka yhdessä luovat syvyytstehosteisen tammi-metsän. Pelin grafiikat ovat pikselimäisiä. Pelikentässä on ääniefektit pelihahmon hyppämiselle ja haukkumiselle, timantin keräämiselle, esteeseen törmäämiselle sekä kuolemalle. Pelin taustalla soi musiikkia. Ääniefektit päätetään tarkemmin pelikehityksen aikana.

Pelissä on muutamia erilaisia valikoita. Päävalikko aukeaa pelaajalle ensimmäisenä pelin käynnistyessä. Päävalikossa soi sama musiikki, kuin pelikentässäkin, ja musiikki jatkuu luontevasti samasta kohdasta peliin siirtyessä, missä se oli valikosta jatkaessa. Valikossa on painikkeet pelin aloittamiselle, pelin ohjauspainikeohjeelle, hahmokaupalle, sekä omien ennätysten tarkasteluun. Valikon painikkeita klikatessa kuuluu ääniefektejä. Pelihahmona toimiva shiba voi myös haukkua. Hahmokaupassa pelaaja voi vaihtaa keräämiään timantteja eri värisiin shiboihin, vaihtoehtoina on punaisen perusversion lisäksi valkoinen sekä mustanrusehtava black & tan-värinen shiba. Kumpikin väri vaihtoehto maksaa 30 timanttia. Kaikki kolme väri vaihtoehtoa ovat päävalikossa animoituina koristeina. Painikeohje-osiossa on kerrottu hyvin yksinkertaisin ja helposti ymmärrettävin tekstein

mikä näppäimistön painike vastaa mistäkin toiminnosta. Toiminnot on esitetty animoidulla shiballa, joka tekee kyseistä toimintoa kuvauksen vieressä. Valikossa ja pelinäköymässä on myös painike, josta pelin äänet saa pois päältä.

Peli pitää kirjaa pelaajan pisteistä, ja listaa kolme parasta tulosta ennätystaulukkoon, jonka pelaaja voi nähdä pelin pysäytysikkunassa. Pysäytysvalikko asettaa nimensä mukaisesti pelin etenemisen tauolle. Pysäytysvalikkoon ohjeistava painike sijaitsee pelikentän vasemmassa alareunassa. Ennätystaulun alapuolella on pelin jatkamis- ja päävalikkoon palaamispainikkeet.

Pelin kohderyhmänä on lapset sekä tasohyppelyistä tai koirista kiinnostuneet aikuiset. Peliä pelataan yksin, ja sen pelaamiseen tarvitaan nettiyhteys, sillä peli julkaistaan ilmaispeleksi Itch.io-pelialustalle, joka toimii verkossa. Peliä ei erikseen markkinoida, eikä sillä tavoitella rahallista tuottoa.

Hauskaa koko perheelle – Ilmainen tasohyppelypeli kaiken ikäisille

Hauku ja hyppää – Monipuoliset toiminnot timanttijahdin aikana

Keräily kannattaa – Kuinka monta erilaista shibaa saat käyttöösi?

Varo vaaraa – monta erilaista vaaratilannetta voi tulla eteen

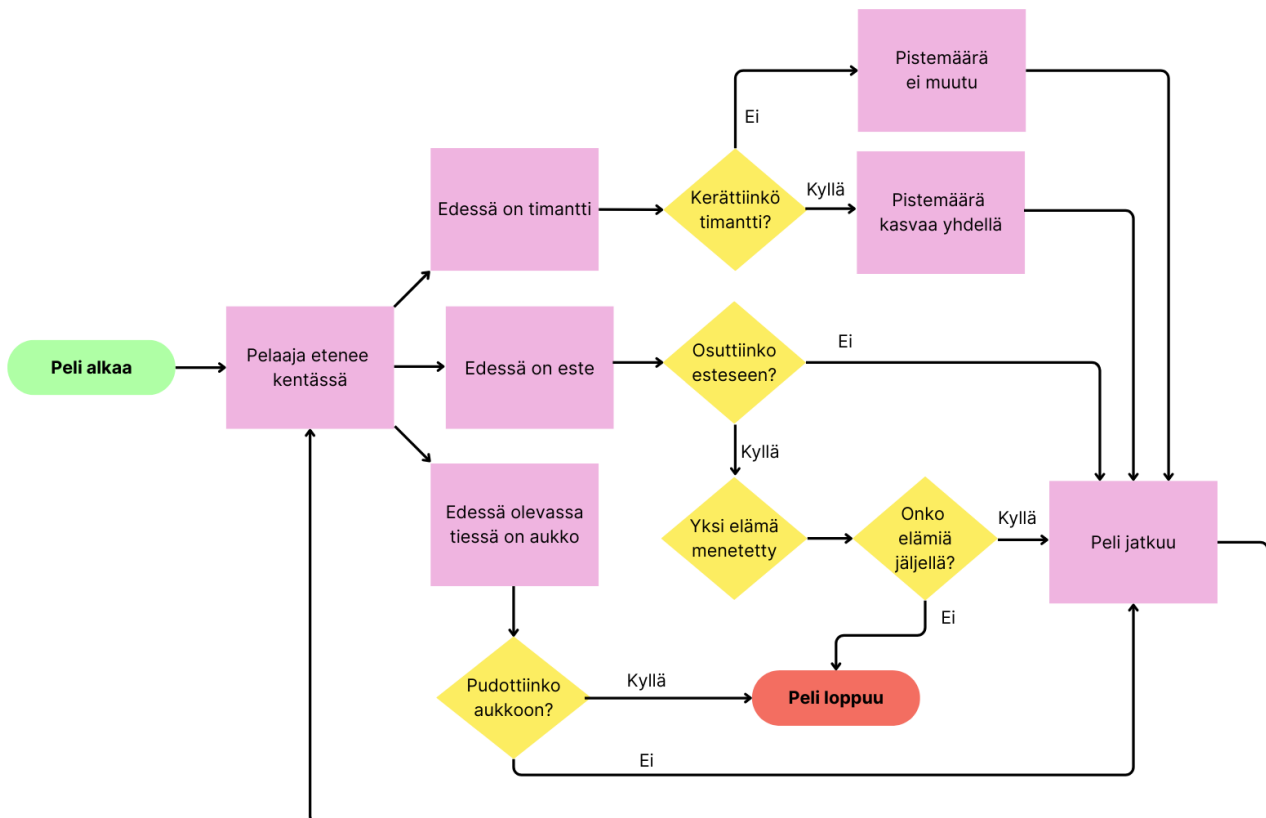
4.2 Vuokaavio

Vuokaavio on visuaalinen työkalu erilaisten prosessien kuvaamiseen, ja se voi huomattavasti selkeyttää laajojen kokonaisuuksien toimintaa. Vuokaaviossa käytetään vakiosymboleita eri osioille: soikiota käytetään prosessin alun ja loppumisen kuvaamiseen, suunnikasta käytetään tulosten kuvaamiseen, suorakulmiot ovat kaikenlaisen toiminnan merkinä ja timanteilla ilmaistaan kahteen vaihtoehtoon jakautuvia päätöksentekokohtia. Symbolien väleissä on nuolia, jotka osoittavat prosessin kulkusuuntaa. (Miro s.a..)

Vuokaavioita on neljä eri tyyppiä: prosessivuokaavio, uimaratavuokaavio, tiedonkulkukaavio sekä työnkulkukaavio. Prosessivuokaavio on etenkin ohjelmistosuunnittelussa ja etenemisen tarkastelussa usein käytetty vaihtoehto. Uimaratavuokaavion avulla voidaan havainnollistaa sellaisia projekteja, joilla ei ole alku- tai loppukohtaa. Tiedonkulkukaavion avulla kommunikaatio-ongelmat eri ryhmien kesken on helpompi selvittää. Työnkulkukaaviota käytetään vakioiksi muodostuneiden toimintatapojen kartoittamiseen. (Miro s.a..)

Shiba & Sparkle-pelin suunnittelun tueksi on laadittu pelissä etenemistä havainnollistava prosessivuokaavio, jossa on kuvattu kaikki pelin etenemiseen vaikuttavat pelitilanteet. Vuokaavio alkaa pelin aloittamisesta, josta on siirtymä pelaajan etenemiseen kentässä. Kentässä edetessä pelaaja voi

kohdata kolme pelin jatkumiseen tai pisteiden keräämiseen vaikuttavaa tapahtumaa: pelaajan edessä voi olla timantti, este tai aukko tiessä. Jos pelaajan edessä on timantti, vuokaavion mukaisesti tarkistetaan, keräsiikö pelaaja timantin. Jos timantti kerättiin, pelaajan keräämä pistemäärä kasvaa yhdellä. Jos timanttia ei kerätty, pistemäärä pysyy ennallaan. Sekä timantin keräämisen, että keräämättä jättämisen jälkeen peli jatkuu normaalisti, kunnes pelaaja tulee taas jonkin kolmen edellä mainitun tapahtuman eteen (kuva 1).



Kuva 1. Pelissä etenemistä kuvaava prosessivuokaavio

Jos pelaajan edessä on este, tehdään tarkistus, osuiko pelaaja esteeseen. Jos osui, pelaaja menettää yhden elämän, jonka jälkeen tarkistetaan, onko elämiä vielä jäljellä. Jos elämät loppuivat törmäyksessä, peli loppuu, mutta jos elämiä on törmäyksen jälkeen vielä jäljellä, peli jatkuu normaalisti seuraavaan vastaan tulevaan tapahtumaan. Mikäli pelaajan eteen tulee tiessä oleva aukko, tarkistetaan, pääseekö pelaaja turvallisesti aukon yli, vai putoaako hän aukkoon. Jos aukko ylitetään turvallisesti, peli jatkuu seuraavaan tapahtumaan saakka, ja jos aukkoon pudotaan, peli loppuu välittömästi (kuva 1).

4.3 Pelin käyttöliittymä

Pelinäkymän käyttöliittymän suunnittelussa on kiinnitetty huomiota sen yksinkertaisuuteen: pelaaja näkee pelihahmon elämät kolmena vaaleanpunaisena sydämenä pelin vasemmassa yläreunassa, ja pelin aikana kerättyjen timanttien lukumäärän pelin oikeassa yläreunassa olevan timanttikuvakkeen vieressä. Pelin aikana soi pirteä musiikki, ja peliä elävöittää pelihahmon hypätessä kuuluva ääniefekti, sekä haukkumisanimaation aikana kuuluva aito koiran haukahdus. Pelaaja voi halutessaan ottaa kaikki pelin äänet pois käytöstä missä tahansa pelinäkymässä, klikkaamalla pelinäkymän vasemmassa alareunassa olevaa kaiutinkuvaketta.

Kun pelaaja saa timantin kerättyä, siitä kuuluu heleä ääni ja piikkimäiseen esteeseen osuessa este räjähtää asiaankuuluvan ääniefektin säestyksellä. Mikäli räjähdys tapahtuu pelihahmon viimeisen elämän aikana, peli päättyy epäonnistumisesta kertovan ääniefektin kera, ja näkyviin tulee pelin päättymisikkuna. Pelin päättymisikkunassa kerrotaan pelaajalle, kuinka monta timanttia hän sai kerättyä, ja kuinka kerättyjen timanttien määrä asettuu ennätyslistassa. Listan alapuolella on painikkeet pelin uudelleen pelaamiseen sekä päävalikkoon palaamiseen. Pelin pysäytysvalikossa on lueteltu pelin ohjaustoiminnot, ja se sisältää myös painikkeet pelin jatkamiseen ja päävalikkoon palaamiseen. Pysäytysvalikossa on animoitu shiba koristeena. Peliin ei tehdä erillistä lopetuspainiketta, sillä peli on tarkoitettu selaimessa pelattavaksi, jonka vuoksi painikkeella ei olisi todellista tarvetta.

Pelin päävalikossa soi sama musiikki, kuin pelin aikana. Valikon keskiosassa on neljä allekkaista painiketta: New Game- painike uuden pelin aloittamiselle, Controls-painike pelin ohjaustoimintojen lukemiseen, Shiba Store-painike pelihahmojen ostamiseen ja käyttöönottoon, sekä Highscores-painike pelaajan omien piste-ennätysten tarkistamiseen. Painikkeiden painamisesta kuuluu äänitehoste. Shiba Storessa pelaajan ostettavissa olevat shibat on listattu vierekkäin pelaajan alkuperäisen pelihahmon viereen, ja jokaisen hahmon alla on painike niiden valitsemiselle. Päävalikon alareunassa on näkyvillä kaikki kolme shiban väri vaihtoehtoa animoituina koirina.

4.4 Peliresurssit

Pelin tekemisessä käytettiin ilmaisia resursseja, lukuun ottamatta pelihahmona käytettävää shibaa. Peliresurssien valinta on tehty vastuullisesti, huomioiden resursseihin liittyvä tekijänoikeus: peliin on valittu vain sellaisia resursseja, joita lupa käyttää ja muokata vapaasti kaikenlaisissa peliprojekteissa. Tärkein peliresurssi oli pelihahmona toimiva shiba, jonka ympärille pelin koko idea rakentui. Pelin taustaksi valikoitui rauhallisen sävyinen, kolmesta kuvatiedostosta koostuva tammimetsikkö. Pelaajan hyppyalustoiden materiaaliksi valittuun peliresurssipakettiin sisältyi alustamateriaalien lisäksi myös erilaisia koriste-esineitä alustoille asetettavaksi, sekä pelin esteenä käytettävä piikikäs

pallo. Kerättävä timantti oli osa isompaa esineresurssipakettia, mutta peliin siitä siirrettiin ainoastaan timantin kuva.

5 Pelin toteutus

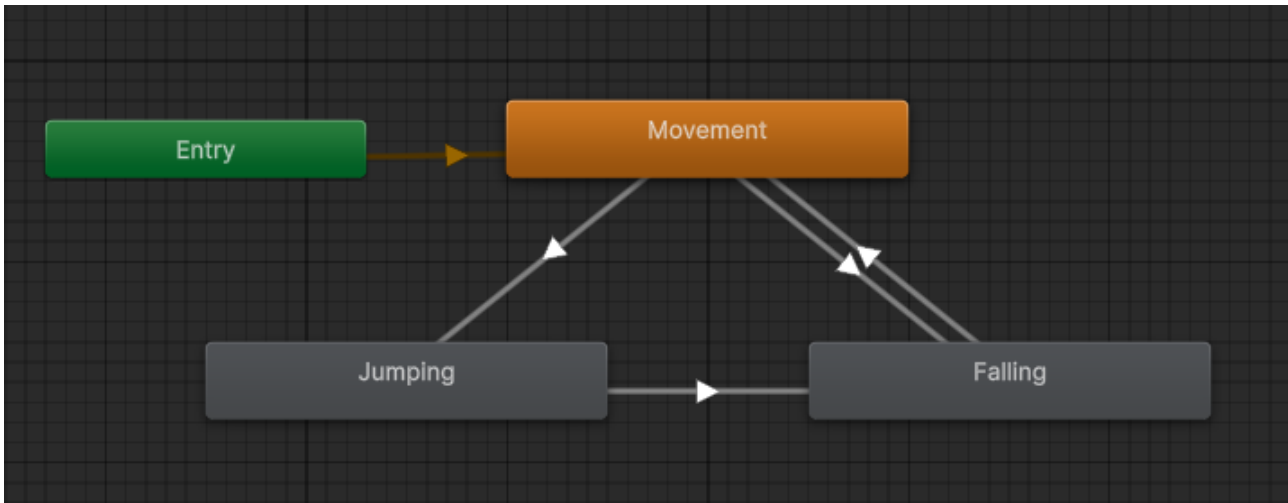
5.1 Hahmon luominen

Aloitin pelin tekemisen pelattavan hahmon luomisella. Hahmon luontipohjana toimi png-kuvatiedosto, jossa on seisovan shiban sprite- eli 2D-kuva. Alkuperäisissä 2D-kuvissa shiba on piirretty seisomaan niin, että sen nenä osoittaa vasemmalle, mutta pelihahmon täytyy edetä oikealle, joten käänsin hahmon ympäri klikkaamalla Unityn Inspector-ikkunassa olevan Flip-osion x-akseliin viittaavaan ruudun aktiiviseksi, jolloin shiba kääntyi katsomaan oikealle päin.

Lisäsin hahmolle kapselin muotoisen Capsule Collider 2D-törmäystunnisteen ja hahmon fysiikoiden noudattamista määrittelevän RigidBody 2D-fysiikkakomponentin. Fysiikkakomponentin avulla pelihahmon liikkeisiin vaikuttaa painovoima, jolloin esimerkiksi reunan yli kävellessä hahmo putoaa alas. Hyppäämisen oikeanlaisen liikeradan varmistamiseksi pelihahmon kiertyminen z-akselin ympäri piti estää, jotta hahmo ei kaadu taakse- tai eteenpäin hypätessä. Hyppääminen on mahdollista vain, jos pelihahmo seisoo alustalla, joten loin peliin väliaikaisen alustan neliömäisestä peliobjektista, jonka päällä pystyin testaamaan hahmon animaatioita. Loin hahmon jalkojen väliin tyhjän peliobjektin, josta peli tunnistaa, koskettavatko hahmon jalat jotain pelin alustoista. Pelin alustoja ja pelaajan törmäyksen tunnistamista varten loin Platform-nimisen kerrostunnisteen. Yhdistin hahmoon pelin ensimmäinen kooditiedoston, jossa määritellään pelaajan liikesuunnat ja hyppyvoima, putoamisen tunnistaminen sekä animaatioiden käynnistämiseen tarvittavien ehtojen toteutuminen. (Biswas 2025a.)

Tein hahmon animaation shiba-resurssipaketin sisältämien 2D-kuvien avulla. Pelihahmon ollessa valittuna, avasin Unityn valikosta animaatioikkunan, johon raahasin haluamani animaation muodostavat 2D-kuvat. Kuvista muodostui animaatioikkunassa olevalle aikajanelle pieniä, timantin muotoisia merkkejä, joita siirtämällä pystyin säätämään animaation toistonopeutta. (Biswas 2025b.)

Pelihahmon liikkumista varten laadin juoksu- ja hyppyanimaatiot, sekä hahmon lepotilana käytettävän animaation, jonka aikana koiran häntä heiluu sen ollessa paikallaan. Animaatioiden vaihtumista varten niiden välille luotiin Unityn animaattorin avulla yhteydet, joissa olevien ehtojen tulee täyttyä, ennen kuin animaatio voidaan toistaa (kuva 2).



Kuva 2. Pelihahmon animaattori

Yhdistin paikallaan seisomisen ja juoksun animaatiot toisiinsa Movement- nimisessä animaatiopuussa, joka mahdollistaa niiden sulavan vaihtumisen keskenään. Ehdoksi seisomisesta juoksuun vaihtamiseen määritin hahmon nopeuden: mikäli nopeus on 0, hahmo seisoo paikallaan, ja mikäli se on yli 1, hahmo lähtee juoksemaan. Hahmon liikkumista määrittävästä animaatiopuusta hyppäämisanimaatioon liitetyllä yhteydellä on kaksi toteutusehtoa, jotta hyppäämisanimaatio voidaan käynnistää: hahmo ei saa koskettaa alustaa, ja sen pystysuuntaisen liikenopeuden tulee olla suurempi kuin 0. Putoamisanimaatio käynnistyy automaattisesti painovoiman vaikutuksesta, kun pelihahmo saavuttaa hyppyvoimansa maksimin, joten ainoana ehtona hyppäämis- ja putoamisanimatioiden välillä on se, että pelihahmo ei saa koskea alustaan. Hahmon palattua alustan päälle, sen animaatioksi muuttuu nopeudesta liikkuen joko paikallaan seisominen, tai juokseminen. (Game Code Library 26.5.2023, 4:00 – 9:32 min..)

Pelihahmon pudotessa juostuaan alustan reunan yli, sen animaatioksi muuttui putoamisanimaation sijaan hyppyanimaatio. Tämän ehkäisemiseksi lisäsin liikkumisen määrittävästä animaatiopuusta yhteyden putoamisanimaatioon, ja laitoin animaation vaihtumiselle kaksi ehtoa: pelihahmo ei saa koskettaa alustaa, ja sen pystysuuntainen liikenopeus on oltava pienempi kuin 0. Tämän avulla hahmon hyppyanimaatiota ei tarvinnut muuttaa, ja hahmon pudotessa animaatioksi muuttuu automaattisesti putoamisanimaatio.

5.2 Kentän luominen

Kentän luomisen aloitin pelin taustakuvan luomisesta. Halusin lisätä peliin elävän oloisen taustan, joka toistuu automaattisesti pelihahmon liikkeessä eteenpäin. Jotta peli näyttäisi siltä, että kenttä oikeasti etenee, lisäsin pelihahmoon Cinemachine-kameran, joka pystyy seuraamaan hahmoa (Game Code Library 12.4.2024, 0:45 – 1:10 min..).

Latasin taustakuvaa varten kolmesta kuvatiedostosta koostuvan peliresurssin. Kuvissa on tam-
mista koostuva metsämaisema, jonka syvyys muodostuu kuvissa olevien puiden koon ja värisävyn
vaihteluilla (kuva 3).



Kuva 3. Pelin taustakuva

Jotta taustakuva muodostuisi oikein, kuville piti lisätä Sorting Layer, eli järjestystaso, joka määrittää sen, mikä kuvista on takimmainen, keskimmäinen ja päällimmäinen. Myös pelihahmolle piti lisätä oma järjestystasonsa, jotta se näkyisi taustakuvan edessä. Tein taustakuvalle oman kooditiedostonsa, jossa määritellään taustakuvan syvyysvaikutelman luovan parallaxEffectin ja kameran luoma etäisyys, sekä kuvan toistuvuus kuvan pituuden mukaisesti. Liitin kooditiedoston jokaiseen kuvaan Unityn Inspector-ikkunassa, ja määritin jokaiselle oman syvyyden: takimmaisena kuvan syvyysarvoksi annoin 0.8, keskimmäiselle 0.6 ja etummaiselle 0.3. Arvot määrittelevät sen, kuinka nopeasti kukin kuva liikkuu kameraan nähden. Etummaisina liikkuu kaikista nopeimmin, ja takimmaisina kaikista hitaimmin. Tämän lisäksi kopioin jokaisen kuvan kahdesti, ja siirsin kopiot kuvan molemmille puolille niin, että puiden kuvat ovat saumattomasti yhdessä. (Game Code Library 12.4.2024, 1:11 – 6:40 min..)

Kuvien toistuvuudessa ilmeni kuitenkin ongelma, joka aiheutti kuvien saumattoman toistumisen katkeamisen, jonka takia kuva ei toistunut halutulla tavalla. Korjasin ongelman muokkaamalla koodia niin, että kuva toistuikin jo kuvan pituuden puolivälissä, täyden pituuden sijaan (kuva 4).

```
using UnityEngine;
// Unity Script (3 asset references) | 0 references
public class BackgroundController : MonoBehaviour
{
    public float startPosition;
    public float length;
    public GameObject cam;
    // The background's movement speed, depending on the camera
    public float parallaxEffect;

    // Unity Message | 0 references
    void Start()
    {
        startPosition = transform.position.x;
        length = GetComponent<SpriteRenderer>().bounds.size.x;
    } // Start

    // Unity Message | 0 references
    private void FixedUpdate()
    {
        float cameraX = cam.transform.position.x;

        // Move background based on camera and parallax effect
        float distance = cameraX * parallaxEffect;
        transform.position = new Vector3(startPosition + distance, transform.position.y, transform.position.z);

        // Endless loop logic
        float temp = cameraX * (1 - parallaxEffect);

        // Fix for looping delay: loop at half-length instead of full
        if (temp > startPosition + length / 2f)
        {
            startPosition += length;
        }
        else if (temp < startPosition - length / 2f)
        {
            startPosition -= length;
        } // if
    } // FixedUpdate
} // class
```

Kuva 4. Taustakuvan koodi

Taustakuvan valmistuttua aloitin pelin tasojen luomisen. Latasin itch.io-sivuston peliresurssikaupasta ilmaisen resurssipaketin, joka sisälsi 2D-kuvan, jossa oli erilaisia ruutukartan osia, joiden avulla voi rakentaa peliin tasoja. Osille oli määriteltä peliresurssin tiedoissa valmis, 32x32 pikselin koko, joten asetin saman koon Unityn Sprite Editor-leikkaustoiminnolle, joka pilkkoi ison ruutukartan yksittäisiksi soluiksi. Leikkaamisen jälkeen tein soluista laattakartan. Laattakartalla voi asettaa peliin soluista muodostuneita laattoja yksi kerrallaan, mutta työn sujuvoittamiseksi tein laattakartalle erillisen sääntölaatan, johon lisäsin yksitellen sääntöjä siitä, miten asetettavan laatan vieressä olevien ruutujen sisältö vaikuttaa laatan ulkonäköön. Esimerkiksi vasemmaksi reunapalaksi asetettavan laatan ulkonäköön vaikuttaa sen sivuilla olevien ruutujen sisältö: laatan vasemmanpuoleisen ruudun on oltava tyhjä, ja oikealla puolella olevassa ruudussa on oltava toinen laatta.

Vastaavanlaisesti esimerkiksi ison alustan keskellä olevan laatan jokaisessa reunassa ja kulmassa on oltava toinen laatta. (Game Code Library 2.6.2023, 0:31 – 4:25 min..)

Loin pelin alkuun kolmesta erillisestä tasosta koostuvan alueen, jossa pelihahmo seisoo odottamassa pelaajan komentoja. Halusin, että jokainen pelikerta olisi erilainen, joten rakensin peliin yhteensä 25 erilaista hyppyalustaa, neljässä eri kategoriassa. Ensimmäinen kategoria sisältää useasta pienemmästä alustasta koostuvan kokonaisuuden. Toisen kategorian alustat sisältävät ramppeja ja korkeita alustoja, joiden yli pelihahmon tulee hyppiä. Kolmas kategoria koostuu suuremmista, hyvin koristeellisista tasoista, joissa on erilaisia maisemia, kuten hautausmaa, pyykin-kuivausnaru sekä pienimuotoinen easter egg- eli peliin piilotettu yllätyselementti, Dark Souls- ja Elden Ring-peleistä tuttu nuotio (Donlan 2024; Rekt Studios 2023). Neljäs kategoria pitää sisällään pieniä, yksittäisiä tasoja. Loin kaikista tasoista erilliset objektimallit niiden käyttöä varten.

Alustojen satunnaistamisen toteuttamisen työvälineenä käytin Object pooling-tekniikkaa, eli peliobjektien uudelleenkäyttämistä varastoinnin kautta. Uudelleenkäyttäminen tekee pelistä helpommin suoritettavan, kun objekteja ei tarvitse jatkuvasti luoda uudestaan tuhoamisen jälkeen. (Unity Technologies s.a. e..)

Aloitin luomalla peliin tyhjän peliobjektin, johon liitin alustojen luomiskoodin. Koodissa määrittelin pelihahmon etäisyys peliin luotavista alustoista. Koodi pitää kirjaa kaikista aktiivisena olevista alustoista, sekä viimeisimpänä luodusta alustasta, estäen sen uudelleen luomisen uudestaan, jotta peli ei toista samaa alustaa useasti peräkkäin. Kun pelaaja on kulkenut alustoista eteenpäin, ne palautuvat varastoon.

Jokaiseen alustan objektimalliin on lisätty End- ja Landing-pisteet, eli loppumis- ja laskeutumispiste. Loppumispiste on nimensä mukaisesti alustan lopussa, ja laskeutumispiste on alustan ensimmäisen tasaisen osion kohdalla, johon pelaaja voi hypätä. Näiden pisteiden välisen korkeuseron kasvaminen liian suureksi kahden peräkkäisen alustan välillä on estetty koodissa, jotta pelaaja voi varmasti hypätä jokaiselle tasolle. Alustojen korkeutta rajoittaa myös pelin näkymä: kamera ei seuraa pelihahmoa pystysuunnassa, joten alustojen luomiselle on asetettu minimi- ja maksimikorkeus (kuva 5).

3 references

```

private void SpawnPlatform()
{
    // Select a random platform prefab
    Transform platform = GetRandomPlatform();
    Transform platformInstance = platformPools[platform].Get();

    // Get platform GameObject points
    Transform landing = platformInstance.Find("Landing");
    Transform end = platformInstance.Find("End");
    Transform top = platformInstance.Find("Top");
    Transform bottom = platformInstance.Find("Bottom");

    if (landing == null || end == null || top == null || bottom == null)
    {
        platformInstance.gameObject.SetActive(false);
        platformPools[platform].ReturnToPool(platformInstance);
        return;
    } // if

    // Get position based on the Landing point
    Vector3 offset = platformInstance.position - landing.position;
    Vector3 alignedPosition = previousEnd + offset;

    // Determine how much vertical space is needed
    float topOffset = top.localPosition.y;
    float bottomOffset = bottom.localPosition.y;

    // Calculate random vertical offset within allowed limits
    float previousY = previousEnd.y;
    float deltaY = UnityEngine.Random.Range(-maxY, maxY);
    float targetY = previousY + deltaY;

    // Make sure the platform stays in the game screen area
    float maxAllowedY = 8f - topOffset;
    float minAllowedY = -8f - bottomOffset;
    targetY = Mathf.Clamp(targetY, minAllowedY, maxAllowedY);

    // Set the platform's position
    platformInstance.position = new Vector3(alignedPosition.x, targetY, 0f);

    // Add the platform to the active platforms list
    spawnedPlatforms.Add(platformInstance);

    // Update the End point reference for spawning next platform
    previousEnd = platformInstance.Find("End").position;
} // SpawnPlatform

```

Kuva 5. Alustan lisäskoodi

Pelissä keräiltävän timanttiobjektin loin objektimalliksi 2D-kuvasta, väisteltävä piikikäs pallo oli valmis peliobjekti alustojen materiaalin sisältäneessä resurssipaketissa. Kumpaankaan objektiin ei ollut saatavilla valmiita animaatioita, joten kirjoitin niille yksinkertaiset koodit, joiden avulla sain luotua peliobjekteihin ylös ja alaspäin liikkuvan animaation. Pelaajan osuessa timanttiin tai piikkiesteseen, objektien animaatio muuttuu: timantin animaatio vaihtuu sähköanimaatioksi, ja piikkiesteen

animaatio muuttuu räjähdysanimaatioksi. Tein animaatiot 2D-kuvista leikkaamalla ne Sprite Editorissa ja siirtämässä leikatut kuvat animointi-ikkunaan, jotta pystyin säätämään niiden toistopeutta. Animaattorin animaatiosiihtymän ehtona timantille on timantin keräys ja piikkiesteelle pelaajan törmäys. Törmäysanimaation jälkeen kumpikin objekti katoaa pelistä.

Alkuperäisen suunnitelmani mukaisesti lisäsin timantit ja piikkiesteet erikseen jokaiseen alustamalliin, mutta huomasin peliä testatessani, että alustojen palatessa varastoon timanttien ja piikkiesteiden animaatiot eivät välttämättä olleet ehtineet suorittaa itseään loppuun asti. Tämän lisäksi peli antoi törmäyksestä kertovan virheentarkastustekstin myös sellaisissa kohdissa, joissa ei näyttänyt olevan objekteja. Poistin erikseen asetetut objektit alustamalleista, ja tein niille oman uudelleenkäyttämiskoodinsa, mutta sen kanssa ilmeni saman kaltaisia ongelmia.

Lopulliseksi ratkaisuksi ongelmaan tein koodin, joka luo objekteja satunnaisesti pelaajan eteen, ja tuhoaa ne, kun pelaaja on edennyt tarpeeksi kauas niistä. Objekteille on jokaisessa alustamallissa tyhjät LowestSpawn ja HighestSpawn-peliobjektit, eli matalimman – ja korkeimman sallitun asettamispaikan objektit. Matalin sallittu asettamispaikka on alustamallin alussa, ja korkein puolestaan alustamallin lopussa. Matalin asettamispaikka on asetettu alustamallissa korkeimmalla olevan tason alueen korkeudelle, jotta timantteja tai piikkiesteitä ei aseteta alustan kanssa samalle tasolle, jolloin ne olisivat pelihahmon ulottumattomissa. Korkeimman asettamispaikan korkeus vaihtelee alustamallista riippuen. Matalimman ja korkeimman asettamispaikan pisteet luovat niin sanottu suorakulmion muotoisen tilan, johon objekteja voi asettua. Objektien ympärille on määritelty niin sanottu ”turvareunus”, eli 3x3 ruudun kokoinen alue, johon ei voi muodostua toista objektia, jotta ne eivät kasaudu liian lähelle toisiaan (kuva 6).



Kuva 6. Piikkiesteet lisäävät haastetta peliin

5.3 Pelinäkömän käyttöliittymä

Peliohjelman käyttöliittymän tärkein osio on pelihahmon elämälaskuri. Hahmolla on pelin alussa kolme elämää, jotka on merkitty pelin vasempaan yläreunaan vaaleanpunaisina sydäminä. Tein sydämet 2D-kuvista objektimalleiksi. Tein kaksi erillistä kooditiedostoa, jotka vaikuttavat elämälaskurin toimintaan. HealthUI-tiedostossa on määritelty sydämiä lisääminen pelinäkömään ja niiden määrän päivittäminen. PlayerHealth-kooditiedosto sisältää piikkiesteeseen osumisen ja siitä seuraavan elämänmenetyksen, sekä elämien alustamisen uutta peliä varten. (Game Code Library 17.11.2023, 0:10 – 5:10 min..) Elämien lisäksi myös pelaajan keräämien timanttien määrä on nähtävissä pelin aikana, pelin oikeassa yläreunassa. Laskurin tunnistaa timanttilaskuriksi sen vieressä olevasta timanttikuvakkeesta. Laskurin määrä kasvaa automaattisesti yhdellä aina, kun pelaaja osuu timanttiin.

Hahmon osuessa kentässä liikkuviin piikkiesteisiin, hän menettää yhden elämän, jolloin yksi vaaleanpunaisista sydämistä muuttuu harmaaksi. Kun kaikki kolme elämää on menetetty, tai kun pelihahmo putoaa alustalta, peli loppuu ja näytölle ilmestyy Game Over-ruutu kertomaan pelaajalle pelin loppumisesta. Lopetusruutua varten tein yksinkertaiset tekstikentät, joista pelaajalle selviää hänen keräämiensä timanttien määrä. Ruudun alareunassa on painike pelin uudelleen pelaamiseen, sekä pelin päävalikkoon palaamiseen. (Game Code Library 24.11.2023, 0:14 – 8:20 min..)

5.4 Valikko ja ponnahdusikkunat

Pelin päävalikkoa varten loin uuden scene- eli kohtausnäkyvän pelitiedoston. Päävalikko sisältää painikkeet pelin käynnistämiseen, ohjaustoimintojen tarkistamiseen, pelihahmokauppaan sekä pelaajan omien ennätystulosten tarkistamiseen. Asetin valikon taustaksi kuvaobjektin, jonka täytin vaaleanpunaisella valikkopeliresurssilla. Ohjeiden tarkistamisen Controls-osioon kirjoitin tekstimuodossa pelihahmon ohjaamiseen käytettävät painikkeet, sekä lisäsin kuvamuodossa nuolinäppäinten kuvakkeet ja pelin pysäytyspainikkeen kuvan. Shiba Store-osioon lisäsin kolme kuvaobjektia, joissa on kolme sinistä kehystä, joissa jokaisen sisällä on kuva pelihahmon mahdollisesta väri vaihtoehdosta. Jokaisen kehyksen alle lisäsin hahmon vaihtopainikkeen. Ennätysten tarkasteluun tarkoitettuun Highscores-osioon lisäsin tekstiobjektit, joihin päivittyä automaattisesti pelaajan senhetkinen kokonaispistemäärä, sekä kolme parasta yksittäisen pelin aikana kerättyä pistemäärää.

5.5 Hahmokauppa

Hahmokaupassa pelaaja pystyy ostamaan pelihahmolle kaksi uutta väri vaihtoehtoa alkuperäisen punaisen värityksen vaihtoehtoiksi. Yksi väri vaihtoehto maksaa 30 timanttia, punainen väri on ilmainen perusväri. Alkuperäinen suunnitelmani hahmon vaihtamiselle perustui hahmomallien luomiseen, ja hahmo-objektin vaihtamiseen pelikoodin avulla. Tästä seurasi kuitenkin lukuisia ongelmia, jotka rikkoivat pelin toimintaa. Lopulta päädyin yrittämään hahmon vaihtamista animaatioihin perustuen: loin jokaiselle väri vaihtoehdolle oman objektimallin, jolle loin ja nimesin animaatiot ja animaattorin toiminnot samalla tavalla, kuin alkuperäiselle pelihahmolle. Tällä tavalla toteutettuna yksi kooditiedosto pystyy ohjaamaan jokaista hahmoa, eikä hahmon yhteys pelin eri objekteihin vaarannu erilliseen objektiin vaihtamisen tarpeesta. Hahmokaupassa oleva valkoinen ja mustanruskea väri vaihtoehto on lukittu pelin alkaessa.

Loin hahmon osto- ja valitsemispainikkeen toimintaa varten kolme kooditiedostoa: CharacterSkin-tiedoston avulla hahmojen animaattorin ja hahmon yksittäisen, alkuperäisen, animaatioissakin käytettävän kuvatiedoston pystyy kytkemään toisiinsa GameCharacterStore-peliobjektissa, johon on kytketty samaa nimeä kantava kooditiedosto, joka muodostaa väri vaihtoehdot listaksi. CharacterStore-koodin avulla puolestaan suoritetaan hahmovaihtoehdon valinta, jonka tueksi loin CharacterSelectButton-koodin, joka sisältää kaikkien hahmonvalinnassa käytettävien painikkeiden toimintaan vaikuttavat koodit. Pelaajan ostaessa tai vaihtaessa pelihahmoa, jokaisen hahmon painikkeen teksti muuttuu sen mukaan, onko kyseinen hahmo ostettu tai pelaajalla aktiivisena (kuva 7).

```

private void UpdateButtonLabel()
{
    // The selected button
    int selectedIndex = PlayerPrefs.GetInt("SelectedCharacterIndex", 0);

    // Change the button text depending if the color is bought or selected
    switch (characterIndex)
    {
        case 0:
            if (redButtonText != null)
                redButtonText.text = (selectedIndex == 0) ? "Selected" : "Select";
            break;

        case 1:
            if (btButtonText != null)
            {
                if (!IsCharacterUnlocked(1))
                    btButtonText.text = "Buy";
                else if (selectedIndex == 1)
                    btButtonText.text = "Selected";
                else
                    btButtonText.text = "Select";
            } // if
            break;

        case 2:
            if (whiteButtonText != null)
            {
                if (!IsCharacterUnlocked(2))
                    whiteButtonText.text = "Buy";
                else if (selectedIndex == 2)
                    whiteButtonText.text = "Selected";
                else
                    whiteButtonText.text = "Select";
            } // if
            break;
    } // switch
} // UpdateButtonLabel
} // class

```

Kuva 7. Hahmokaupan painikkeiden tekstinmuutoskoodi

5.6 Äänet

Lisäsin peliin taustamusiikin, joka jatkuu saumattomasti riippumatta siitä, onko pelaaja valikossa tai pelissä, tai siirtymävaiheessa niiden välillä (Rehope Games 25.3.2023, 1:05 – 3:25 min.). Taustamusiikin lisäksi etsin itch.ion resurssikaupasta äänitehosteita pelin erilaisille toiminnoille, kuten hahmon hyppäämiselle, timantteihin tai piikkiesteisiin törmäämiselle sekä painikkeiden painamiselle. Suurimman osan äänitehosteista pystyin määrittelemään samaan MusicController-tiedostoon taustamusiikin lisäksi, ja kytkemään ne Unityn Inspector-ikkunassa niihin liittyviin peliobjekteihin. Jotkin pelin painikkeista ovat käyttöliittymässä erilaisten objektien, kuten pelin päättymisikkunan, sisäisinä. Jotta tämän kaltaisiin painikkeisiin saisi äänitehosteen toimimaan, loin erillisen Button-Sound-kooditiedoston, jossa on painikkeen painalluksen äänitehosteen soittamisen koodi. Yhdistin koodin painikkeen onClick-toimintoon. Lisäsin pelin valikkoon ja pelinäkömään myös äänien

vaimentamispainikkeen, johon liitin yksinkertaisen kooditiedoston, joka hiljentää AudioListener-rakennusosan äänenvoimakkuuden.

6 Pelin testaus ja julkaisu

Peliä ei voi tehdä kokeilematta itse sen toimintaa, mutta kuten Kempainen (2019, 124) toteaa, on kriittisen palautteen saaminen osin haastavaa pelintekijän omalla testauksella, joten pelin testaamista suoritettiin myös ulkopuolisten toimesta. Testiryhmäksi valikoitui joukko eri ikäisiä aikuisia, joista osa harrastaa pelien pelaamista, ja osa ei ole pelannut lainkaan, tai hyvin vähäisesti. Testiryhmän pelejä pelaamattomat jäsenet valikoituivat testaajiksi, vaikka he eivät kuulukaan pelin kohderyhmään, sillä heiltä voi saada rakentavaa palautetta sellaisista pelin ominaisuuksista, jotka kokeneemmat pelaajat saattavat sivuuttaa (Kempainen 2019, 126).

Kaikki testaajat ovat henkilökohtaisesti minulle tuttuja, mikä on otettava huomioon testituloksia tutkiessa. Vaikka testaajille on painotettu, että vastausten täytyy olla täysin rehellisiä, eivätkä he saa ajatella vastauksiensa vaikutusta minuun henkilökohtaisesti, on aina riski, että testihenkilöiden tunteminen vaikuttaa testitulosten paikkansapitävyyteen, sillä he saattavat tietoisesti tai tiedostamattaan pyrkiä miellyttämään testituloksissaan (Abbott 2022.)

6.1 Ensimmäinen testaus

Pelin ensimmäisen testauksen pääaiheena oli pelihahmon ohjattavuus ja animaatioiden toiminta, sekä kenttään ilmestyvien tasojen ja esineiden arviointi. Kerroin testaajille, että peli on loputtomasti jatkuvaa tasohyppelyä, jotta he eivät turhaudu kentän loppumista odottaessa. Testaajille jaettu peliversio sisälsi peliin suunnitelluista valikoista ainoastaan pelin päättymisikkunan, johon oli liitetty pelin uudelleenkäynnistyspainike. Testiversio ei myöskään sisältänyt ääniefektejä.

Latasin testattavan peliversioon itch.io-nettisivustolle pakattuna Zip-tiedostona, ja jaoin linkin peliin sähköpostitse testihenkilöille. Sähköpostiin oli myös liitetty linkki Webropol-sivustolla tehtyyn testauskyselyyn, jossa oli 16 pelin toimintoihin liittyvää kysymystä. Osa kysymyksistä oli avoimia kysymyksiä, ja osassa oli valinta-asteikko 1–5 välillä (liite 1). Kysely toimi täysin anonyymisti, eli pelaajilta ei tiedusteltu yhteystietoja, heidän ikäänsä tai pelikokemustaan.

Ensimmäiseen kyselyyn vastasi 5 henkilöä. Yksi henkilö oli aloittanut kyselyyn vastaamisen, mutta ei ollut lähettänyt vastaustaan. Testaajat kokivat hahmon ohjaamisen helpoksi: vastaukset jakautuivat arvosanojen 4 ja 5 välille. Pelihahmon animaatioiden vaihtumiseen testaajat suhtautuivat hieinan laajemmalla skaalalla, vastausten vaihdellessa arvosanojen 3 ja 5 välillä, 60 % oli vastannut arvosanaksi 4. Arvosanat vaihtelivat 4 ja 5 välillä myös tasojen välillä hyppelyn helppoudesta kysyttäessä. Alustojen välinen etäisyys jakoi vastaajien mielipiteitä laajemmin arvosana-asteikon eri ääripäihin: 50 % vastaajista antoi alustojen etäisyydelle arvosanaksi 3, ja loput vastaukset jakautuivat tasan 25 % sekä arvosanalle 2, että arvosanalle 4.

Peliobjekteista kysyttäessä timanttien kerääminen koettiin helpoksi arvosanoilla 4 ja 5. Piikkiesteiden väistely koettiin huomattavasti hankalammaksi: arvosanat vaihtelivat 1 ja 3 välillä, keskiarvona oli 2. Timanttien ja piikkiesteiden koko koettiin yleisesti hyväksi, mutta eräs testaaja oli antanut palautetta, että joissain tilanteissa piikkiesteet saattoivat olla liian lähellä toisiaan, jonka vuoksi niitä oli vaikea väistää. Pelinäkömään yläreunassa olevien elämä- ja timanttilaskurien koko koettiin keskinkertaiseksi, vastausten keskiarvona oli 3.

Testaajien ensivaikutelmat pelistä olivat positiivisia. Pelin estetiikkaa kommentoitiin mukavaksi ja värimaailmaa levolliseksi. Peli koettiin yleisesti hyväksi perustasohyppelypeliksi. Pelin kiinnostavuus jakautui arvosanojen 2 ja 4 välille, keskiarvona 3,4. Haastavista kohdista kysyttäessä useampi vastaaja kirjoitti piikkiesteiden haasteellisuudesta: aiemmin mainitun liian lähekkäin olemisen lisäksi testaajat olivat myös kiinnittäneet huomiota siihen, että jotkin piikkiesteet muodostuivat hankaliin paikkoihin pelihahmon eteen, vaikeuttaen tasojen välillä liikkumista. Eräs testaaja ehdotti peliin helpompaa vaikeustasoa, jossa ei olisi piikkiesteitä ollenkaan. Liian helpoksi testaajat kokivat timanttien keräilyä, eräs testaaja ehdotti timanttien piilottamista esineiden taakse, tai hankalammin tavoitettaviin paikkoihin. Yksi testaajista koki koko pelin liian helpoksi, pohtien pelin kohdeyleisön olevan lähinnä lapsia. Eräs testaajista puolestaan oli sitä mieltä, että pelissä täytyykin olla helpompia kohtia, että mielenkiinto pysyy yllä.

Pelin visuaalisuuteen liittyvään kysymykseen tuli sekä positiivisia, että negatiivisia vastauksia. Osa testaajista kehui pelin taustaa ja värimaailmaa, ja pelin koettiin olevan nostalginen ja harmoninen. Osa testaajista kirjoitti, että pelin visuaalisuus vaatii vielä hiomista, ja yksi testaaja olisi odottanut pelissä olevan aidomman oloista grafiikkaa. Pelin uudelleenkäynnistämisen testaaajilla ei ollut ongelmia. Testaajat löysivät pelistä virhetilanteita: timantit ja piikkiesteet ilmestyivät joissain kohdissa pelaajan ollessa melkein niiden kohdalla. Muutama testaaja myös huomasi, että pelihahmo saattoi pudotessaan jäädä alustan reunaa vasten jumiin.

Testaajat kokivat epäselväksi hahmon ohjaamisen testauksensa alussa, sillä ensimmäisessä testiversiossa ei ollut ohjeistusta hahmon liikuttamisesta. Yksi testaaja oli pohtinut kentän alkuosiossa olevaa rakennelmaa, joka on tarkoitettu estämään pelaajan liikkumista kentässä taaksepäin. Testaaja oli pohtinut, onko rakennelmaa pitkin tarkoitus kiipeillä, tai onko sinne piilotettu jotain. Yksi testaaja oli myös pohtinut mikä timanttien keräämisen tarkoitus on, sillä ensimmäisessä testiversiossa ei ollut pelihahmon vaihtamistoimintoa, jota varten timanteja kerätään.

6.2 Toinen testaus

Pelin toisen testauksen aiheena oli peliin lisätyt ääniefektit ja taustamusiikki, valikoiden toiminta ja pelihahmon vaihtaminen. Testaukseen liittyvässä kyselyssä oli 13 kysymystä (liite 2). Testi oli

toteutettu anonymisti, samoin kuin edellinenkin testaus. Testauskyselyyn vastasi neljä henkeä. Testaajista 100 % piti valikon toimintoja erittäin selkeinä. Hahmon vaihtaminen oli koettu myös selkeäksi, sillä vastaukset jakautuivat tasan puoliksi arvosanojen 4 ja 5 välillä.

Valikon kokoon ja ulkonäköön testaajat olivat olleet tyytyväisiä, valikon kokoa kommentoitiin juuri sopivaksi, ja tekstien fonttikokoa hyvin luettavaksi. Yksi testaajista kirjoitti olevansa tyytyväinen siihen, että pelin pysäytysvalikossa lukee pelin ohjaustoiminnot. Ohjaustoimintojen opastus koettiin selkeiksi ja riittäviksi. Pelin äänien vaimentamisen- ja pysäyttämisen painikkeet koettiin myös sopivan kokoisiksi ja helposti löydettäviksi. Pysäyttämispainikkeen toiminnassa ei ilmennyt ongelmia.

Ääniefektit ja taustamusiikin toistuvuus toimivat testaajilla hyvin. Ainoa äänitehosteisiin liittyvä kritiikki, mitä testaajilta tuli, liittyi pelihahmon kuoleman äänitehosteesta, joka oli muita äänitehosteita äänekkäämpi. Yksi testaajista kirjoitti kyselylomakkeeseen myös huomionsa, että kuoleman äänitehoste soi suuremmalla äänenvoimakkuudella pelihahmon pudotessa rotkoon, kuin piikkiesteeseen osumisen aiheuttaman kuoleman aikana. Yksi testaajista oli kirjoittanut, että äänenvaimennuspainike ei toimi pelin loppumisikkunan ollessa näkyvässä, mutta en pystynyt toistamaan kyseistä havaintoa itse kokeilemalla, joten en tiedä oliko kyseessä peliin vai itch.io-sivustoon liittyvä ongelmatilanne. Äänitehosteisiin ja taustamusiikkiin oltiin tyytyväisiä, testaajien mielestä äänet sopivat pelin teemaan hyvin.

Ongelmakohtista kysyttäessä yksi testaajista kirjoitti, että piikkiesteet hankaloittavat etenemistä vielä tässäkin pelin versiossa. Tarkennusta piikkien haastavaan sijaintiin ei ollut kuitenkaan kirjoitettu. Toinen testaaja kirjoitti, että pelihahmo jäi yhdellä pelikerralla pelikentän alussa jumiin, eikä hän pystynyt liikuttamaan pelihahmoa roikkumisen aikana, tämän aiheutti hyppäystoiminnon rajaaminen maassa seisomiseen. Yhdessä tasossa oli myös näkymätön esine, jonka päälle hahmo jäi seisomaan. Kolmas testaaja kirjoitti, että pelihahmo menee välillä tasoissa olevien esineiden taakse piiloon. Tämä on yksi pelin ominaisuuksista, ja se ei onneksi kuitenkaan häirinnyt asiasta kirjoittanutta testaajaa, vaan hän vertasi toimintoa Super Mario 3-peliin, jossa on samankaltainen toiminto.

Kehityskohteista kysyessä testaajat kirjoittivat, että peliin voisi lisätä ryömintätoiminnon pelihahmolle. Äänitehosteiden ja musiikin äänenvoimakkuuden säätämistä myös muuten, kuin kokonaan vaimentamalla, oli myös toivottu. Hahmon vaihtaminen kesken pelin, sekä mahdollisuus tuhota piikkiesteitä oli myös mainittu. Peliin kaivattiin lisäksi taustatarinaa pelihahmon timanttien keräämiselle ja toisten shibojen ”vapauttamiselle”.

6.3 Testaustulosten perusteella tehdyt muutokset peliin

Ensimmäisen testin perusteella muokkasin peliin ilmestyvien timanttien ja piikkiesteiden muodostumisalueita siirtämällä jokaisesta alustamallista objektien alinta ja ylintä ilmestymispaikkaa määrittävät objektit kesemmälle, jotta objekteja ei voi ilmestyä sellaisiin paikkoihin, joissa ne tekisivät pelissä etenemisestä mahdotonta. Ajan rajallisuuden vuoksi en keskittynyt timanttien ja piikkiesteiden muodostumispaikkojen ja -etäisyyden täydelliseen uudelleen kehittämiseen, vaan siirsin asian jatkokkehitykseen.

Toisen pelitestauksen aikana huomattu näkymätön objekti osoittautui käyttämättä jääneeksi timantin muodostuspaikaksi hyppytasojen ja esineiden alkuperäisestä, yhdessä muodostamisen yrityksestä. Jokaisesta tasomallista löytyikin neljä käyttämätöntä peliobjektia piikkiesteiden sekä timanttien muodostumiselle, jotka olin unohtanut poistaa muodostamistapaa vaihtaessa. Muokkasin myös kuoleman äänitehosteen voimakkuutta 50 % hiljaisemmaksi.

6.4 Pelin julkaisu

Pelin valmistuttua julkaisin sen **itch.io**-verkkosivulle, joka on markkinapaikka digitaalisen materiaalin kehittäjille. Sivustolla pystyy ostamaan ja pelaamaan pelejä, sekä lataamaan erilaisia peliresursseja. Sivusto on etenkin itsenäisesti pelejä kehittävien käyttäjien suosiossa. (The itch.io team s.a..)

Shiba & Sparkle-peli piti rakentaa WebGL-muodossa, sillä peli on tarkoitettu vain verkkoselaimessa pelattavaksi, pelin lataaminen itch.io-sivustolta pelaajan tietokoneelle on estetty. WebGL-muotoon rakennettaessa pelin rakennuskansioon muodostuu pelin lisäksi index.html-tiedosto, joka mahdollistaa pelin toiminnan verkkoalustoilla (Unity Technologies s.a.f.). Pelin rakennuttua pakkasin muodostuneet tiedostot Zip-tiedostoksi ja latasin ne itch.io-sivustolle. Asetin pelin julkiseksi, ja lisäsin sille lopuksi kuvakkeen.

7 Pohdinta

Tämän opinnäytetyön tavoitteena oli kehittää julkaisukelpoinen tasohyppely-peli. Työhön liittyi neljä tutkimuskysymystä: minkälaisia ohjelmistoja pelin kehittämisessä tarvitaan, mitä pelisuunnittelussa tulee ottaa huomioon, kuinka resurssien rajallisuus vaikuttaa pelin kehittämiseen, sekä miten pelin testauttaminen vaikuttaa pelin kehitykseen?

Vastaus ensimmäiseen tutkimuskysymykseen rajautuu käsittelemään tämän opinnäytetyön tuotoksena olevan pelin kehittämisessä käytettyihin ohjelmistoihin ja työkaluihin. Tärkein työkaluista on pelimoottori Unity, jolla Shiba & Sparkle-peli kehitettiin. Vuonna 2005 ensijulkaisunsa saanut Unity on yksi maailman johtavista pelimoottoreista. Shiba & Sparklen kehittämisessä hyödynnettiin Unityn ilmaista tilausmallia. Unityn peliresurssikaupalle ei ollut tuotoksen kehityksessä tarvetta. Pelin koodit kirjoitettiin Microsoftin Visual Studio-tekstieditorin ilmaisversiolla. Visual Studio on Unityn virallinen suositeltu editori, jossa koodi kirjoitetaan C#-, eli C Sharp-ohjelmointikielellä, joka on pelin eri osiin orientoitunut ohjelmointikieli, jonka avulla pelin saa jaettua pienempiin kokonaisuuksiin. Pelikehityksen eri vaiheet ladattiin versionhallintatyökalu GitHubin pilvipalveluun GitHub Desktop-sovelluksen kautta. Versionhallinnalla pystyttiin turvaamaan pelikehityksen joustavuus, sillä muuttaman kerran pelin koodit aiheuttivat isoja ongelmia, jotka sain ratkaistua vain lataamalla aiemman version pelistä, ja aloittamaan ongelmakohdan kehitystyön uudestaan.

Toinen tutkimuskysymys oli: mitä pelisuunnittelussa tulee ottaa huomioon. Pelisuunnittelija, tai suu-remmissa yrityksissä pelisuunnittelijat, laativat pelistä pelikonseptin, joka on tarkka kuvaus pelistä ja siihen liittyvistä osa-alueista, kuten pelin ulkonäöstä, pelin kohderyhmästä sekä liiketoimintamallista. Pelisuunnittelijan vastuualue voi olla yksittäinen osa pelistä, tai laajempi kokonaisuus. Käyttöliittymää suunnitellessa pelisuunnittelijan tulee ottaa huomioon pelaajalle syötettävän informaation määrä: liian suuri, kerralla pelaajalle syötettävä informaatiomäärä voi pahimmillaan kääntää pelaajan kiinnostuksen pois pelistä. Informaatio on keskeisessä osassa myös pelisuunnittelijan työskentelyssä osana tiimiä, sillä pelisuunnittelijan tulee osata kommunikoida tiimin jäsenten kanssa, ja osata ottaa vastaan myös negatiivista palautetta. Peliin saattaa tulla yllättäviäkin muutoksia, joten omiin, tai muiden, esittämiin ideoihin ei ole kannattavaa kiintyä liiallisesti.

Ajankäyttö ja aikatauluttaminen ovat pelisuunnittelijan työssä tärkeitä sekä pelin suunnittelun, että toteuttamisen kannalta. Huolellisella aikataulutuksella ja suunnittelutyöllä voidaan taata pelin kriittisten osioiden valmistuminen ajallaan. Pelialusta, jolla peliä pelataan, tulee myös ottaa huomioon peliä suunnitellessa, sillä eri alustoilla on omat ohjauspainikkeensa ja grafiikkarajallisuutensa. Myös pelin pelityyli ja -kategoria vaikuttavat pelin suunnittelutyöhön. Pelisuunnittelijalla tulee olla hyvä kokonais käsitys pelin kehityksen kaikista vaiheista.

Kolmas tutkimuskysymys, kuinka resurssien rajallisuus vaikuttaa pelin kehittämiseen, oli yksi pelikehityksen aikana eniten vaikuttanut asia. Shiba & Sparkle-peliä suunnitellessa minulla oli visio, että pelin hahmona on shiba. Tämän vision ympärille rakentui kaikki muu pelin sisältö. En ole taiteellinen henkilö, joten en halunnut luoda itse pelissä käytettäviä peliresursseja, vaan etsin niitä peliresurssikaupoista. Pyrin pitämään pelin kehitystyön ilmaisena, mutta se osoittautui mahdottomaksi, osin omien vaatimusteni, ja toisaalta myös resurssitarjonnan kannalta. Resurssikaupoissa olevat ilmaiset shiba-peliresurssit eivät olleet mielestäni tarpeeksi miellyttäviä, joten päädyin ostamaan hieman yli 2 euroa maksavan shiba-peliresurssin pelin päähahmoksi. Muut resurssit löytyivät ilmaiseksi, mutta niiden etsimiseen ja miellyttävimpien vaihtoehtojen valitsemiseen kului paljon aikaa. Erityisesti äänitehosteita ja taustamusiikkia etsiessä peliin sopivia vaihtoehtoja ei ollut kovin helppoa löytää. Suuri osa resurssikaupoista ei myöskään tarjonnut mahdollisuutta kuunnella äänitehosteita ilman, että ne latsin omalle tietokoneelle, joka puolestaan lisäsi resurssien etsimiseen kulunutta aikaa. Pahimmassa tapauksessa, mikäli peliä varten on tarkat vaatimukset, resurssien rajallisuus voi lisätä pelikehitykseen kuluvaan aikaa, mikäli resursseja täytyy luoda itse alusta alkaen. Pelin budjetti voi myös kasvaa huomattavasti, mikäli resursseja joudutaan ostamaan ulkopuolisilta toimijoilta.

Neljäs, ja viimeinen, tutkimuskysymys miten pelin testauttaminen vaikuttaa pelin kehitykseen, sai konkreettisen vastauksen Shiba & Sparkle-pelin testauttamisen aikana. Peliä testanneen testiryhmän jäsenet ovat keskenään eri ikäisiä, osa heistä pelaa pelejä harrastuksenaan ja osa ei pelaa ollenkaan. Testiryhmä oli jaettu erilaisiin pelaajiin sen vuoksi, että heiltä saataisiin mahdollisimman monipuolisia kommentteja heidän erilaisten kokemustensa ja näkökulmiensa perusteella. Testijäseniä oli 8 henkeä, mutta kaikki eivät vastanneet kyselyihin. Ensimmäiseen kyselyyn vastasi 5 henkeä, toiseen 4. Testaajat löysivät pelistä virheitä, jotka olivat minulta jääneet huomaamatta. Heiltä tuli myös hyviä kehitysideoita pelin jatkokehitystä varten. Testauksessa tulee usein ilmi, mitkä osat pelissä ovat pelaajille epäselkeitä tai vaikeita, vaikka ne olisivatkin pelin kehittäjällä itsestäänselvyksiä, tästä esimerkkinä ensimmäisestä testiversiosta puuttunut pelihahmon ohjauspainikkeiden ohjeistus, josta melkein jokainen testaaja kysyi minulta erikseen. Testaajilta voi tosin olla myös haastavaa saada selkeitä vastauksia testikysymyksiin, avoimeen kysymykseen vastaukseksi tuleva "ok" ei ole välttämättä kovin hyödyllinen pelikehityksen kannalta.

Opinnäytetyön tavoite, julkaisukelpoisen pelin kehittäminen, täyttyi. Tuotoksena oleva peli ei kuitenkaan sisällä kaikkea alun perin suunnittelemani sisältöä, esimerkiksi aloitusvalikkoon, ohjeistussivulle, sekä pysäytysvalikkoon suunnittelemani animoidut shibat jäivät valikon ja pysäytysikkunan lopullisen ulkonäön puitteissa pois pelin toteutuksesta. Suunnittelemani koiran haukkumistoinnin sisällyttäminen peliin jäi jatkokehitykseen aikataulullisten haasteiden vuoksi, vaikka olinkin jo ehtinyt äänittää oman koirani haukahduksia pelissä käytettäväksi. Jatkokehityksessä koiran

haukkumiseen on mahdollista yhdistää testauksessa esiin noussut piikkiesteiden tuhoaminen: koira voisi haukkumisellaan tuhota edessä olevia, hankalasti ohitettavia piikkiesteitä. Muita jatkokehitysideoita, joita testauksessa tuli hyvin esille, ovat hahmon vaihtaminen pelikentässä pelaamisen aikana, sekä taustamusiikin ja äänitehosteiden äänenvoimakkuuksien muokkausmahdollisuus. Lisäksi pelihahmon jumittuminen alustojen reunaan pudotessaan jäi aikataulullisista syistä jatkokehitykseen. Testauksessa esille noussut ryömintätoiminto koiralle ei ole suoranaisesti kovin helppo toteuttaa, sillä shiba-peliresurssiin ei sisällynyt erillistä ryömintäanimaation mahdollisuutta, ainoa samankaltainen mahdollisuus olisi tehdä makuuanimaatio koirasta, mutta sen tassut eivät liiku animaatioon käytettävissä kuvatiedostoissa, joten koira lähinnä liukuisi maata myöten. Timanttien ja piikkiesteiden sijoittelu on yksi tärkeimmistä jatkokehityksen aiheista, sillä nykyinen muodostumistapa ei ole pelaajan kannalta kovin reilu. Käyttöliittymää voisi myös kehittää hieman esteettisesti miellyttävämmäksi jatkossa.

Opinnäytetyötä tehdessä opin paljon uutta pelikehityksestä. Aiempi kokemukseni ei sisältänyt pelihahmon vaihtamista toiseen, eikä loputtomasti toistuvan taustan käyttöä, ja ne olivatkin kenties eniten aikaa vieviä asioita pelin kehittämisessä. Koen saaneeni lisää itsevarmuutta peliohjelmointiin tuotosta tehdessäni. Ongelmatilanteita oli lukuisia, mutta jokaiseen löytyi lopulta toimiva ratkaisu. Koen myös ymmärtäväni Unityn erilaisia toimintoja aiempaa selkeämmin, ja uskon tästä olevan hyötyä työelämässä.

Opinnäytetyön tekeminen sujui hyvin vaihtelevasti. Teoriaosuuden kokoaminen ja materiaalien tutkiminen sujui nopeasti, mutta pelin kehitystyö vei huomattavasti oletettua enemmän aikaa. Oma, alkuperäinen aikatauluni venyi huomattavasti opinnäytetyöprosessin aikana. Olen kuitenkin tyytyväinen opinnäytetyön tuotokseen, etenkin, kun olen pelikehityksessä vielä aloittelijan tasolla. Pelin tekeminen oli stressaavaa, mutta ongelmakohtien selvittäminen ja toimivien ratkaisujen löytäminen oli palkitsevaa.

Shiba & Sparkle-peli on pelattavissa itch.io- sivustolla, linkki peliin: <https://sallarikr.itch.io/shiba-sparkle>.

Lähteet

Abbott, R. 2022. Family (Mis)fortunes – How your research participants might be biasing your findings. Luettavissa: <https://medium.com/hippo-digital/family-mis-fortunes-how-your-research-participants-might-be-biasing-your-findings-9d6b5eed6f9c>. Luettu: 11.5.2025.

ACG Technical Editors Team 8.6.2023. What is the C# programming language?. Pluralsightin blogi. Luettavissa: <https://www.pluralsight.com/resources/blog/cloud/what-is-the-c-programming-language>. Luettu: 27.4.2025.

Ansbacher, C. 25.3.2025. How to Identify Your Game’s Ideal Target Audience. Solstenin blogi. Luettavissa: <https://solsten.io/blog/identify-your-target-audience>. Luettu: 28.4.2025.

Biswas, R. 2025a. O’Reilly Learning. Creating a 2D Platformer Player Controller. Intranet. Katsottu: 26.4.2025.

Biswas, R. 2025b. O’Reilly Learning. Adding Player Animations. Intranet. Katsottu: 26.4.2025.

Catness Game Studios 12.12.2024. All about video game types, consoles, and platforms. Catness Game Studiosin blogi. Luettavissa: <https://catnessgames.com/blog/video-games-types-consoles-platforms/>. Luettu: 28.4.2025.

Cohen-Peckham, E. 2019. How Unity built the world’s most popular game engine. Luettavissa: <https://techcrunch.com/2019/10/17/how-unity-built-the-worlds-most-popular-game-engine/>. Luettu: 26.4.2025.

Designveloper 15.10.2024. What is C#: Definitions, Strengths & Usages. Designveloperin blogi. Luettavissa: <https://www.designveloper.com/blog/what-is-c-sharp/>. Luettu: 27.4.2025.

Donlan, C. 2024. Bonfires are still my favourite FromSoftware idea. Luettavissa: <https://www.euro-gamer.net/bonfires-are-still-my-favourite-fromsoftware-idea>. Luettu: 9.5.2025.

Evans, C. 2.6.2023. Learn These 10 Unity Game Development Languages By 2023. CodeX. Blogi. Luettavissa: <https://medium.com/codex/learn-these-10-unity-game-development-languages-by-2023-31c3d66c66e8>. Luettu: 26.4.2025.

Game Code Library 2.6.2023. Easy Tilemaps and Dynamic Auto Tiling – Unity 2D. Video. Katsottavissa: <https://www.youtube.com/watch?v=8UctaO5DwUE>. Katsottu: 5.5.2025.

Game Code Library 24.11.2023. Game Over Screen and Retry Button – 2D Platformer Unity #15. Video. Katsottavissa: <https://www.youtube.com/watch?v=6x8aCzUuIDw>. Katsottu: 11.5.2025.

- Game Code Library 26.5.2023. Idle, Run and Jump Animations – Platformer Unity 2D. Video. Katsottavissa: https://www.youtube.com/watch?v=Sg_w8hlp4Y. Katsottu: 30.4.2025.
- Game Code Library 12.4.2024. Infinite Parallax Scrolling Background – Unity 2D Complete Tutorial. Video. Katsottavissa: <https://www.youtube.com/watch?v=AoRBZh6HvIk>. Katsottu: 4.5.2025.
- Game Code Library 17.11.2023. Player Health and On Hit Effect – 2D Platformer Unity #14. Video. Katsottavissa: <https://www.youtube.com/watch?v=zmQ3hKWxsmE>. Katsottu: 11.5.2025.
- GeeksForGeeks 2023. Introduction to Visual Studio. Luettavissa: <https://www.geeksforgeeks.org/introduction-to-visual-studio/>. Luettu: 27.4.2025.
- GitHub Inc. s.a. Getting started with GitHub Desktop. Luettavissa: <https://docs.github.com/en/desktop/overview/getting-started-with-github-desktop>. Luettu: 27.4.2025.
- GitLab Inc. s.a. What is version control?. Luettavissa: <https://about.gitlab.com/topics/version-control/>. Luettu: 27.4.2025.
- Hardman, C. 2024. Game Programming with Unity and C#: A Complete Beginner's Guide. 2. uudistettu painos. Apress. New York. E-kirja. Luettu: 24.4.2025.
- Kempainen, J. 2019. Pelisuunnittelijan peruskirja. Aviador Kustannus. Helsinki. Luettu: 22.4.2025.
- Kramarzewski, A., De Nucci, E. 2023. Practical Game Design. 2. uudistettu painos. Packt Publishing. Birmingham. E-kirja. Luettu: 23.4.2025.
- Lanzinger, F. 2021. 2D Game Development with Unity. CRC Press. Boca Raton. E-kirja. Luettu: 24.4.2025.
- MasterClass 2021. Learn About Platform Game: 7 Examples of Platform Games. Luettavissa: <https://www.masterclass.com/articles/platform-game-explained>. Luettu: 12.5.2025.
- McKenzie, C. 2024. Git vs. GitHub: What's the difference?. Luettavissa: <https://www.theserverside.com/video/Git-vs-GitHub-What-is-the-difference-between-them>. Luettu: 27.4.2025.
- Microsoft s.a. Compare Visual Studio 2022 Editions. Luettavissa: <https://visualstudio.microsoft.com/vs/compare/>. Luettu: 27.4.2025.
- Minor, J. 2025. The Best Video Game Subscription Services for 2025. Luettavissa: <https://uk.pcmag.com/games/143824/the-best-video-game-subscription-services>. Luettu: 28.4.2025.

Miro s.a. Mikä on vuokaavio?. Luettavissa: <https://miro.com/fi/flowchart/what-is-a-flowchart/>. Luettu: 28.4.2025.

Pavlovic, D. 2020. Video Game Genres: Everything You Need to Know. Luettavissa: <https://www.hp.com/us-en/shop/tech-takes/video-game-genres>. Luettu: 28.4.2025.

Radigan, D. s.a. What is a branching strategy?. Luettavissa: <https://www.atlassian.com/agile/software-development/branching>. Luettu: 27.4.2025.

Rehope Games 25.3.2023. Uncover the Simple Trick to KEEP MUSIC PLAYING Between Scenes! Video. Katsottavissa: <https://www.youtube.com/watch?v=xswEpNpucZQ>. Katsottu: 14.5.2025.

Rekt Studios 2023. Easter Eggs: The Beating Heart of an Industry. Luettavissa: <https://www.linkedin.com/pulse/easter-eggs-beating-heart-industry-rekt-studios/>. Luettu: 9.5.2025.

The itch.io team s.a. About itch.io. Luettavissa: <https://itch.io/docs/general/about>. Luettu: 21.5.2025.

Trang, H. 7.7.2024. What is a programming language? Exploring popular programming languages. Tokyo Tech Labin blogi. Luettavissa: <https://tokyotechlab.com/blogs/ngon-ngu-lap-trinh>. Luettu: 12.5.2025.

Trang, H. 3.7.2024. What is game programming?. Common languages in game development. Tokyo Tech Labin blogi. Luettavissa: <https://tokyotechlab.com/blogs/lap-trinh-game-la-gi>. Luettu: 27.4.2025.

Unity Technologies 2010. Unity Technologies Launches 3rd Party Marketplace 'Unity Asset Store'. Luettavissa: <https://unity.com/news/unity-technologies-launches-3rd-party-marketplace-unity-asset-store>. Luettu: 26.4.2025.

Unity Technologies s.a. a. Plans and pricing. Luettavissa: <https://unity.com/products>. Luettu: 26.4.2025.

Unity Technologies s.a. b. Unity's Asset Store. Luettavissa: <https://docs.unity3d.com/Manual/AssetStore.html>. Luettu: 26.4.2025.

Unity Technologies s.a. c. Asset Store. Luettavissa: <https://assetstore.unity.com/>. Luettu: 26.4.2025.

Unity Technologies s.a. d. Game Object. Luettavissa: <https://docs.unity3d.com/Manual/class-GameObject.html>. Luettu: 27.4.2025.

Unity Technologies s.a. e. Introduction to Object Pooling. Luettavissa: <https://learn.unity.com/tutorial/introduction-to-object-pooling#>. Luettu: 10.5.2025.

Unity Technologies s.a.f. Create and publish WebGL builds. Luettavissa: <https://learn.unity.com/tutorial/create-and-publish-webgl-builds#>. Luettu: 21.5.2025.

Xsolla 8.12.2023. Play smarter: Boost your game's success with targeted audiences. Xsollan blogi. Luettavissa: <https://xsolla.com/blog/play-smarter-boost-your-games-success-with-targeted-audiences>. Luettu: 28.4.2025.

Liitteet

Liite 1. Ensimmäisen pelitestauksen testauskysymykset

1. Kuinka helpoksi koit hahmon ohjaamisen?
1 = Erittäin vaikeaa, 5 = Erittäin helppoa
2. Kuinka sulavia hahmon animaatioiden vaihtumiskohdat olivat?
1 = Erittäin tönkköjä, 5 = Erittäin sulavia
3. Kuinka helpoksi koit alustojen välillä hyppimisen?
1 = Erittäin vaikeaa, 5 = Erittäin helppoa
4. Kuinka koit alustojen välisen välimatkan?
1 = Alustat olivat liian kaukana toisistaan, 5 = Alustat olivat liian lähellä toisiaan
5. Millaiseksi koit timanttien keräämisen?
1 = Erittäin vaikeaa, 5 = Erittäin helppoa
6. Millaiseksi koit piikkien väistelyn?
1 = Todella vaikeaa, 5 = Erittäin helppoa
7. Mitä mieltä olit timanttien ja piikkien koosta?
8. Mitä mieltä olit elämien ja kerättyjen timanttien laskurin koosta?
1 = Liian pienet, 5 = Liian suuret
9. Millaisen ensivaikutelman sait pelistä?
10. Oliko pelissä jotain kohtia, mitkä koit liian haastavaksi?
11. Oliko pelissä jotain kohtia, jotka koit liian helpoksi?
12. Mitä mieltä olit pelin visuaalisuudesta?
13. Toimiko pelin uudelleen käynnistys sujuvasti?
14. Kuinka mielenkiintoiseksi koit pelin?
15. Tuliko peliä pelatessa virhetilanteita?
16. Oliko pelissä jotain epäselvää?

Liite 2. Toisen pelitestauksen testauskysymykset

1. Kuinka selkeitä valikon toiminnot olivat?
1 = Erittäin epäselkeitä, 5 = Erittäin selkeitä
2. Kuinka helppoa hahmon vaihtaminen oli?
1 = Erittäin hankalaa, 5 = Erittäin helppoa
3. Mitä mieltä olit valikon koosta ja ulkonäöstä?
4. Mitä mieltä olit äänien vaimentamis- ja pelin pysäytyspainikkeiden koosta ja ulkonäöstä?
5. Oliko taustamusiikin toistuvuudessa ongelmia?
6. Oliko ääniefektien toiminnassa ongelmia?
7. Oliko äänien hiljentämispainikkeen toiminnassa ongelmia?
8. Oliko pelin pysäyttämispainikkeen / Esc-näppäimen toiminnassa ongelmia?
9. Oliko pelin ohjaustoimintojen opastus selkeä?
10. Mitä mieltä olit taustamusiikista ja ääniefekteistä?
11. Oliko pelissä vielä jotain epäselvää?
12. Tuliko pelissä vastaan ongelmakohtia?
13. Löytyikö pelistä kehitettäviä kohteita?