



Terveyttä ja hyvinvointia edistävä oppimispeli

– Case Munuais- ja maksaliitto

Lauri Kara, Arttu Mikkanen, Gabriela Salonen

2023 Laurea



Laurea-ammattikorkeakoulu

**Terveyttä ja hyvinvointia edistävä oppimispeli
– Case Munuais- ja maksaliitto**

Lauri Kara
Arttu Mikkonen
Gabriela Salonen
Tietojenkäsittely, Restonomi
Opinnäytetyö
Marraskuu, 2023

Toiminnallisen opinnäytetyön tavoitteena oli kehittää yläkouluikäisille suunnattu oppimispeli. Terveellisten elämäntapavalintojen tekeminen ja päihteidenkäytön välttäminen ovat pelin keskeisiä teemoja. Opetuksellinen sekä kasvatuksellinen tehtävä toteutuu tarinallista valintapeliä pelaten.

Pelin luontivaiheessa ohjelmointikieliksi valittiin HTML, CSS ja JavaScript. Pelinluontialustoja ei käytetty. Pelillistäminen liittyy yleisesti palveluiden kehittämiseen. Palvelutuotetta kehitettiin asiakaslähtöisesti palvelumuotoilun ja yhteiskehittämisen menetelmiä käyttäen osaksi Munuais- ja maksaliiton digitaalisia palveluita.

Vuoden kestänyt projekti oli monialainen ja se toteutettiin yhdessä opiskelijoiden, toimeksiantajan, graafikon, äänisuunnittelijan, kohderyhmän edustajien, sidosryhmien sekä muiden yhteistyökumppaneiden kanssa. Valmis tuote luovutettiin julkaisutilaisuudessa Munuais- ja maksaliitolle ja järjestö sai siihen kaikki oikeudet.

Bachelor's Degree Programme in Business Information Technology, Hospitality Management & Service Design

(BHD)

Lauri Kara, Arttu Mikkanen, Gabriela Salonen

An Educational Game that Promotes Health and Well-Being – A Case Study of Kidney and Liver Association

Year

2023

Pages

88

The goal of this functional thesis was to develop a learning game aimed at middle schoolers. Making healthy lifestyle choices and avoiding substance abuse are central themes of the game and an educational task, which is realized by playing a story-based choice game.

HTML, CSS and JavaScript were chosen as programming languages during the game's creation phase. No game creation platforms have been used. Gamification is generally related to the development of services. The service product was developed to be customer-oriented using service design and co-development methods as part of the Finnish Kidney and Liver Association's digital services.

The year-long project was multidisciplinary and was implemented together with students, the client, a graphic designer, a sound designer, representatives of the target group, stakeholders, and other co-operation partners. The finished product was handed over to the Finnish Kidney and Liver Association at the publication event and the organization got all the rights to it.

Keywords: service design, learning game, programming, health

Sisälllys

1	Johdanto.....	7
2	Toimeksiantaja ja toimintaympäristö	8
3	Projektin taustat.....	9
3.1	Projektiorganisaation rakenne	11
3.2	Tavoitteet	12
3.3	Menetelmät	12
3.4	Projektin tiedostot ja niiden luovutus asiakkaalle.....	14
3.5	Aikataulu.....	15
4	Taustatutkimus.....	16
4.1	Benchmarking.....	16
4.2	Empatiakartta	17
5	Määrittely	19
5.1	Oppimispeli.....	19
5.2	Pisteytys	21
6	Pelin ideointi ja suunnittelu.....	22
6.1	Ideointi.....	23
6.2	Prototyyppi.....	25
6.3	Prototyypin tarina	27
6.4	Prototyypin ohjelmointi	28
7	Ohjelman logiikka	29
7.1	Olio-ohjelmoinnin toteuttaminen pelissä	30
7.2	Korttiluokat	31
7.3	Funktio changeStats.....	36
7.4	Funktion draw toiminta.....	38
7.5	Funktio drawCard	38
7.6	Funktio drawInCard.....	39
7.7	Funktiot callAnimation ja drawAnimations.....	40
7.8	Funktiot changeChapter ja changeKuvat	42
7.9	Funktio die	44
7.10	Funktio drawResultScreen	46
7.11	Funktio restartGame	48
7.12	HTML-canvakselle piirtäminen	49
8	Pelin tuottaminen	50
8.1	Tarina ja hahmokonsepti.....	50
8.2	Tekijänoikeudet.....	52

8.3	Käsikirjoitus ja luvut	53
8.4	Pelin markkinointi ja testaus	57
8.5	Tiedonhaku ohjelmointia varten	62
8.6	Ohjelmointikielet	62
8.7	Komentointi koodissa	64
8.8	Äänisuunnittelu	64
8.9	Pelin viimeistely	65
8.10	Lisäominaisuudet peliin palautteen pohjalta	66
8.11	Koodin viimeistely	66
9	Bugien korjaaminen	66
9.1	Korttiolioiden sisältö	67
9.2	Pelin uudelleenkäynnistyksen ongelmat	68
9.3	Animaatiobugi	68
9.4	Kuvien näkyvyysongelma	69
9.5	Konsolin käyttö bugien korjaamisessa	71
9.6	Ongelmat pelin kuvaruudun skaalautumiseen liittyen	71
10	Yhteenveto ja arviointi	72
	Lähteet	75
	Kuviot	78
	Kuvat	78
	Taulukot	80

1 Johdanto

Nuoruus on elämäntarinan alkua, mikä on kokeilemisen ja kasvun aikaa. Teini-iässä vapaus lisääntyy ja myös vastuu omasta elämästä kasvaa. Päätöksiä tehdään itse ja yhdessä muiden ihmisten kanssa. Tietoa hyvien ja terveellisten valintojen tueksi on saatavilla runsaasti ja sen on oltava nuorten saatavilla myös heille sopivalla tavalla. Palveluita kehitettäessä tulee tuntee ja ymmärtää eri asiakasryhmien tarpeita. Nuoret osallistuvat entistä enemmän heille kohdennettujen palveluiden suunnitteluun. Nykyajan toimintaympäristöt ovat fyysisten ympäristöjen lisäksi myös digitaalisia, joissa tärkeitä ovat mm. erilaiset sosiaaliset verkostot, viestintä ja interaktiivisuus. Turvallisuusuhat verkossa ja sosiaalisessa mediassa ovat myös todellisia. Uudet tekniikat ja menetelmät sekä tekoäly määrittelevät uudelleen sen, mitä on mahdollista tapahtua.

Palvelumuotoilu on asiakaslähtöinen tapa suunnitella ja toteuttaa palveluita. Palveluiden ja tuotteiden suunnitteluun sekä kehittämiseen tarkoitettujen menetelmien käyttäminen auttaa kehittämään asiakasymmärrystä tavalla, joka mahdollistaa lopputuotoksen sopivuuden ja parhaan mahdollisen käyttäjäkokemuksen palvelun kohderyhmälle. Pelien käyttö kouluttamis- ja oppimistarkoituksessa on levinnyt hyvin yleisesti palvelualoille ja työelämään. Pelikonseptin tarkoituksenmukaisuus ja sopivuus sille tarkoitettuun toimintaan edellyttää perehtymistä tuotteen loppukäyttäjii unohtamatta kuitenkaan pelin sisältöä, jonka välittyminen ymmärrettävästi pelaajalle on kuitenkin oppimispelin varsinainen tehtävä. Sisällön on määritellyt tämän työn toimeksiantajan eli tilaajan lisäksi myös sidosryhmät.

Pelin ohjelmoinnissa on käytetty HTML, CSS ja JavaScript -ohjelmointikieliä. Sivun ulkoasu ja tyylit on tehty HTML:llä ja CSS:llä, mutta peli itsessään on ohjelmoitu kokonaan JavaScriptillä. Pelikokemukseen ja pelistä oppimiseen vaikuttavat mm. pelaamiskertojen määrä, moniaistillisuuden toteutuminen, pelaajan motivointi palkitsemalla, pelin vaikeustaso, tarinallisuus, hahmot sekä dialogi. Onnistunut ohjelmointi luo pelin perusrungon, joka mahdollistaa pelin toiminnallisuuden. Pelissä on noin 3000 riviä koodia.

Pelin tarinassa seikkailevat piirroshahmot ja valintoja tekevä pelaaja. Tarinallisen pelin luominen monialaisessa tiimissä organisoidusti on vaatinut paljon toimivaa viestintää ja myös oman alan asiantuntemuksen lisäksi muihin aloihin perehtymistä. Työhön on kuulunut asiakkaan määrittelemien sisältöjen hakua ja tuottamista, ideointia, tarinankerrontaa, ohjelmointia ja visuaalista sekä äänimaailman suunnittelua kuten kuvakäsikirjoituksen tekemistä. Pelin koodi on ladattuna GitHub-alustalle ja se sisältää eri luokkia ja funktioita, jotka mahdollistavat valintapelin toiminnot ja funktiot. Yrityksistä huolimatta kaikkien opiskelijatiimin

jäsenten perehdyttäminen ohjelmointiin ei tuottanut kaivattua tulosta ja tiimin sisällä työtehtävät jakautuivat hyvin omien koulutusalojen, osaamisen sekä kiinnostuksen kohteiden mukaisesti.

2 Toimeksiantaja ja toimintaympäristö

”Järjestöt tarjoavat eri ikäryhmille tietoa ja taitoja terveellisten elintapavalintojen tueksi. Järjestötoiminnassa korostuvat esimerkiksi matalan kynnyksen liikuntapalvelut, terveellistä ravitsemusta edistävä toiminta ja terveyden edistämishankkeet” (THL 2022).

Munuaisilla on monia tärkeitä tehtäviä. Ne poistavat nestettä ja kuona-aineita kehosta ja säätelevät verenpainetta. Munuaiset säätelevät punasolujen muodostumista ja tuottavat D-vitamiinia. Maksa on ihmiskehon suurin sisäelin. Maksa sijaitsee ylävatsaontelossa pallean alla, ja sitä suojaavat kylkiluut. Terveen aikuisen maksa painaa 1-1,5 kg. Maksalla on keskeinen rooli aineenvaihdunnassa. Maksalla on yli 500 toimintoa. Et pärjää ilman maksaa. (Munuais- ja maksaliitto 2022.)

Kenellä tahansa voi olla munuais- tai maksasairaus. Ne vaikuttavat tavalla tai toisella tuhansiin suomalaisiin. Munuais- ja maksayhdistys tukee potilaiden, elinsiirtojen saajien ja heidän läheistensä hyvinvointia ja oikeuksia. Yhdistys tiedottaa munuais- ja maksasairauksista ja niiden ehkäisystä, edistää elinluovutusta ja vaikuttaa yhteiskunnallisesti. Liittoon kuuluu 18 jäsenyhdistystä, joissa on yhteensä yli 6 000 yksittäistä jäsentä. Jäsenyhdistyksessä voit tavata muita, jotka käyvät läpi samaa. Yhdistyksen jäseneksi voivat liittyä kaikki tästä toiminnasta kiinnostuneet. Liittyminen on paras tapa tukea liittojen ja yhdistysten tavoitteita ja toimintaa. Munuais- ja maksaliitto ry:llä on keskustuimisto Helsingissä ja aluetoimistot Tampereella, Kuopiossa, Jyväskylässä ja Oulussa. (Munuais- ja maksaliitto 2022.)

Munuais- ja maksaliitto tekee työtä arjessa tapahtuvien parempien elintapavalintojen edistämiseksi ja tarjoaa palveluillaan tietoa terveyden riskitekijöiden tunnistamiseen ja ehkäisyyn. Yhdistys arvioi munuais- ja maksasairauksien lisääntyvän tulevaisuudessa ja näkee järjestön tarjoamien palvelutarpeiden kasvavan. (Munuais- ja maksaliitto 2022.)

Laadukkaat digitaaliset palvelut kannustavat oppijoita toimimaan ennakoivasti, työskentelemään tavoitteidensa eteen ja olemaan vuorovaikutuksessa muiden kanssa (Laatumäärittely 2021, Opetushallitus). Munuais- ja maksaliitto mainitsee viimeisimmässä strategiassaan munuais- ja maksasairauksia ja niiden ennaltaehkäisyä koskevan tiedon tarjoamisen osana tehtäväkenttäänsä. Lisäksi yhdistys toimii yhteiskunnallisena vaikuttajana (Munuais- ja maksaliitto 2022).

Hyvinvointimme on monikerroksinen ja siihen vaikuttavat eri ryhmät, joissa elämme. Hyvinvointimme yksilönä, ryhmänä, organisaationa, yhteiskuntana ja osana maailmaa. (Toim. Rahmel, TARU-oppimateriaalipankki, 31.)

Toimeksiantajan sekä opiskelijoiden lisäksi projektissa oli mukana asiantuntijoita Helsingin kaupungilta, Irti Huumeista ja Ehyt -järjestöistä. Ehyt pyrkii toimimaan vuorovaikutteisesti keskustellen nuorten kanssa yhdessä kriittisesti pohtimaan minkälaisissa tilanteissa esim. päihteet ja niiden käyttö tulevat esille (Ehyt 2022).

Irti Huumeista ry tarjoaa valtakunnallisia vertais- ja ammattitukipalveluita huumeita käyttävien läheisille, toipuville ja huumeita käyttäville. Toiminnan tarkoituksena on vähentää ja ehkäistä huumeiden käytöstä aiheutuvia haittoja sekä yksilölle että yhteiskunnalle. (Irti huumeista ry, 2022.)

Kaikki tietävät, että huumeet ovat epäterveellisiä ja ennen kaikkea vaarallisia, keskivertoylempästä astelainen kykenisi monesti antamaan osuvamman valistuksen kuin koulun tekstikirja. Epätietoinen huumevalistus, joka ajaa epäloogisilla syytöksillä huumeiden vaaroja on nykyään terveystiedon tunneilla suurinta huvitusta nuorille. Kuten kieltolain aikaan Yhdysvalloissa, se mikä on kiellettyä voi innostaa sen kokeiluun. Huumeet kehittyvät nopeammin kuin lainsäädäntö tai huumevalistus aiheista, mutta edes hetkellisesti saamme Suomen markkinoille pelin, joka on ajan tasalla huumeista ja niiden vaaroista. Myös joitakin muista maailman vaaroista on otettu huomioon pelissä.

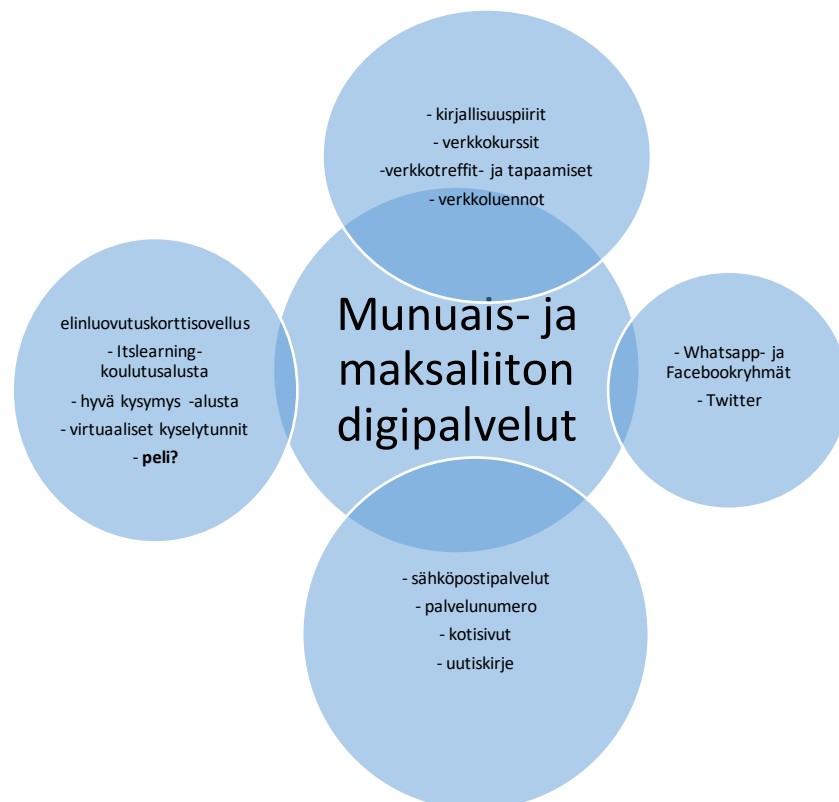
3 Projektin taustat

Keväällä 2022 Laurean opiskelijat toteuttivat Munuais- ja maksaliiton jäsenille, sidosryhmille ja henkilökunnalle osana käyttäjäkeskeisen suunnittelun opintokokonaisuutta kyselyn koskien liiton digitaalisia palveluita. Kysely oli osa liiton digitaalisten palveluiden kehittämistä. Tässä yhteydessä opiskelijoilta nousi esiin ajatus opetuspelistä osana liiton digitaalisia palveluita. Pelin kohderyhmäksi toimeksiantaja määritteli yläkouluikäiset lapset ja nuoret. Pelin tarkoitus olisi antaa nuorille tietoa maksan terveyteen liittyvistä haitoista ja tukea nuorta tekemään valintoja oman hyvinvointinsa edistämiseksi. Peli sisältäisi tietoa mm. terveellisestä ruokavaliosta sekä päihteiden haitoista. Peli ohjelmoitaisiin alusta alkaen itse.

Pienet lapset voivat syntyä tai myöhemmin sairastua munuais- tai maksasairauteen ja joutua elinsiirtoon ja lapsen sairaus vaikuttaa koko perheeseen. Vanhemmat ovat vastuussa monimutkaisesta hoidosta ja lääkityksestä kotona. Kokemusten jakaminen muiden samaa kokeneiden perheiden kanssa auttaa. Nuoruusiässä nuoret alkavat miettiä terveyttään uudestaan. He voivat alkaa miettiä sairauden kulkua, aiempia hoitoja ja niiden vaikutuksia elämäänsä.

Sairauden mahdolliset rajoitukset vaikuttavat osaltaan siihen, miten tulevaisuutta suunnitellaan ja miten arjessa jaksetaan. (Munuais- ja maksaliitto 2023.)

Pienten lasten kasvatus on perinteisesti liitetty tarinankerrontaan. Tarinankerronnan voimaa ja mahdollisuuksia nuorten ja aikuisten maailmassa ei useinkaan tunnisteta. Fenomenologinen kasvatus ja tutkimus narratiivisella ajattelulla, juonenkehityksellä ja jännitteillä rikastuttaa oppimista huomattavasti. Tarinat vahvistavat merkityksiä ja niillä on uskomattoman myönteinen ja oppimista edistävä vaikutus. (toim. Rahmel 2019, 2021, 10) ”Parhaimmillaan tarina näkyy, tuntuu, maistuu, tuoksuu ja kuuluu (tarinakone.fi 2022).”



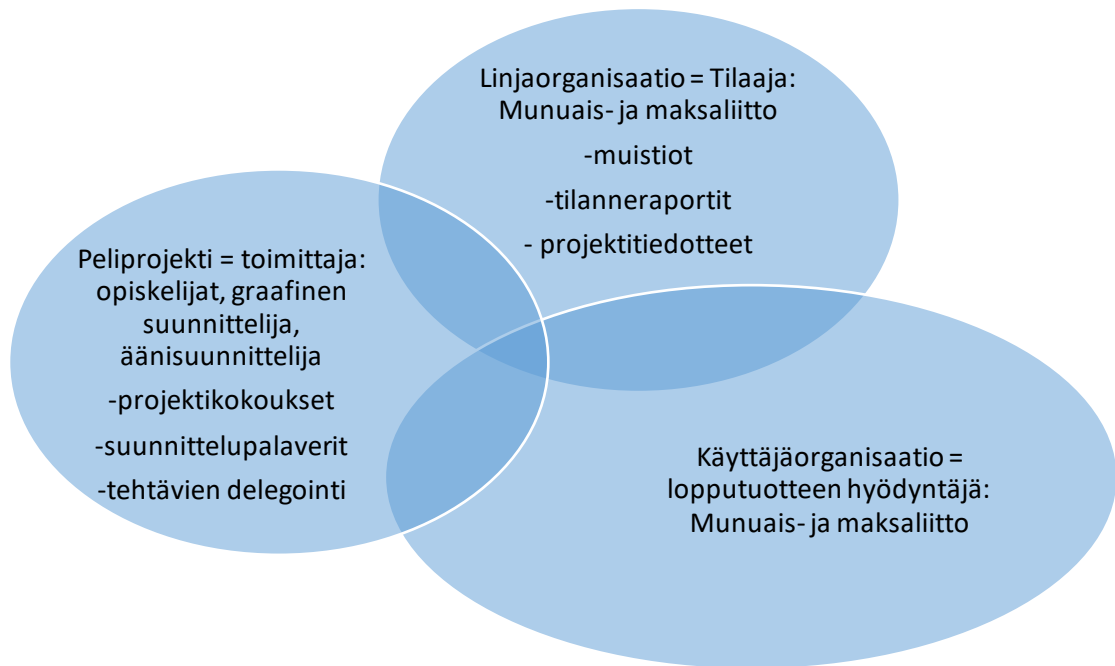
Kuvio 1: Munuais- ja maksaliiton digitaaliset palvelut peliprojektin alussa

Voi hyvin syin olettaa, että palvelualallakin aletaan yhä laajemmin näkemään erilaisia pelillistämiprojekteja. Palvelualan asiakaslähtöisyys on pelillisyyden näkökulmasta mielenkiintoinen osa-alue. (Hamari, 2017.)

Palvelualalla pelillistämispotentiaalia löytyy professori Juho Hamarin mukaan esimerkiksi asiakas kohtaamistilanteista. Tuotekehitystyössä yritysten asiakkaiden ja henkilöstön merkitys on avainasemassa, sillä heidän ovat ideoijia ja osajia. (Raatikainen 2008, 60.)

3.1 Projektioorganisaation rakenne

Organisaatio on väline asetettujen tavoitteiden saavuttamiseksi. Toisin sanoen hankkeen organisaatioon vaikuttavat aina sen toimeksiannon laajuus ja luonne sekä hankkeen ja perusorganisaation välinen suhde. (Ruuska, 2012, 128.)



Kuvio 2: Projektin suhde ympäristöön

Idea uuden palvelutuotteen kehittämiseen oli lähtöisin opiskelijoilta. Tutkimustoiminnan kautta voidaan löytää uusia palvelutarpeita ja voidaan käynnistää uusia tuotekehittelyn projekteja. (Raatikainen 2008, 61.)

Käyttäjiä on helppo ottaa mukaan tuotekehitykseen innovoimaan uusia ja erottuvia tuotteita, heiltä ja markkinoilta on saatavilla hyvin tietoa (Bergström, Leppänen, 2016, 185). Lähestymällä kohderyhmää ja käyttäjiä uudella luovalla konseptilla, joka voi olla tuote, palvelu tai näiden kahden yhdistelmä, voidaan ratkaista ongelma vaikkapa siitä, kuinka saada nuoret kiinnostumaan terveydestä ja hyvinvoinnista (Reaktori 2022).

Terveysalan pelejä voidaan jaotella monin tavoin. Terveysalan pelejä suunnataan käyttötarkoituksesta riippuen sekä ammattilaisille, opiskelijoille että suurelle yleisölle. Suurelle yleisölle tarkoitetut pelit voivat olla ns. kasvatuksellisia terveyspelejä ja sisältää tietoa vaikkapa terveellisestä ravitsemuksesta, hyvinvoinnista ja sairauksien välttämisestä. Digitaaliset oppimispelit voivat olla virtuaalisia simulaatioita tai ne voivat olla konsolipelejä ja/tai tietokoneella pelattavia videopelejä. (Pelikasvattajan käsikirja 2023, 95.)

3.2 Tavoitteet

Toiminnallisen opinnäytetyön tavoitteena on synnyttää toiminnallinen tuotos. Yleensä ratkaistaan jokin ongelma, kuvaillaan jokin prosessi ja analysoidaan sen vaiheita tai kehitetään jokin alan käytäntöä. Palvelua tai tuotetta rakennetaan, kokeillaan ja kehitellään käytännön toiminnan kautta. Toiminnallinen opinnäytetyö -nimityksen rinnalla käytetään synonyymisesti myös monimuotoista opinnäytetyötä. (Metropolia 2022.)

Tässä toiminnallisessa työssä oli tavoitteena tuottaa peli toimeksiantajalle palvelumuotoilun menetelmiä hyödyntäen monialaisesti yhteistyössä eri alojen sekä sidosryhmien asiantuntijoiden, osaaajien ja suunnittelijoiden kanssa. Työssä käytetään pelistä sekä käsitettä oppimispeli että valintapeli, koska sitä voidaan käyttää myös vaikkapa koulussa opetuksellisesti oppitunneilla ja/ tai järjestöjen tiloissa ajanvietteenä. Sen on oltava helposti saatavilla ja sen käyttämisen on oltava sujuvaa. Käyttäjäkokemuksen/ystävällisyyden huomioonottaminen on osa tätä työtä.

”Halutaan, että nuoret lähtevät pelaamaan tätä peliä eivätkä kompastu jo alkumetreillä siihen, että peli ei näy tai toimi tarkoituksenmukaisesti (Munuais- ja maksaliitto 2023).” Esimerkiksi skaalaukseen keskityttiin pelin testausvaiheessa, kun huomattiin siihen liittyvät ongelmat. Pelin koko muuttuu näytön koon mukaan dynaamisesti, joka tekee pelattavuudesta parempaa. Nettisivujen skaalautuminen on osa kokonaisuutta, joka tunnetaan myös nimellä responsiivinen verkkosuunnittelu, mikä tarkoittaa, että verkkosivut toimivat hyvin erilaisissa laitteissa ja eri kokoisilla näytöillä (Dude 2019). Responsiivisuus on erittäin tärkeää sekä käytettävyyden että hakukonenäkyvyyden kannalta (eLuotsi 2023). Pelikokemus muodostuu ympäristöllisistä ja yksilöllisistä tekijöistä. Pelistä tekee koukuttavan positiiviset asiat kuten haastavuus, visuaalisuus, kiinnostavat tarinat sekä vuorovaikutteisuus (Pelikasvattajan käsikirja 2023, 101). Pelisuunnittelun tavoitteet määritellään usein dynamiikan tai estetiikan kautta (Hämäläinen, Takatalo, 2017). Projektin aloituskokouksessa 24.8.2022 pelin tavoitteiksi määriteltiin, että pelin tulisi olla houkutteleva ja jopa koukuttava, seikkailullinen sekä valistava. Lisää pelin sisällöllisistä tavoitteista kerrotaan tämän raportin myöhemmissä luvuissa sekä liitteessä 3.

3.3 Menetelmät

Palvelumuotoilu on menetelmä, jolla tuotetaan palveluita ja tuotteita asiakaslähtöisesti. Raportti etenee palvelumuotoilun Tuplatimanttimallin mallin mukaisesti sen yleisyyden ja tuttuuden vuoksi. Siitä oli melko helposti materiaalia saatavilla mm. opintojen muista vaiheista. Samalla menetelmällä oli toteutettu aiempia kurssitöitä. Palvelumuotoilu on monialaista ja poikkitieteellistä. Palvelumuotoilun ensimmäisessä ja toisessa vaiheessa pyritään

rakentamaan asiakasymmärrystä ja kahdessa seuraavassa vaiheessa keskitytään itse palvelutuotteen kehittämiseen ja konseptointiin. Käytännössä kyseessä on kuitenkin iteratiivinen prosessi, mikä tarkoittaa sitä, että projekti ei välttämättä etene lineaarisesti eteenpäin vaan siinä palataan takaisin mm. testauksen kautta aiempiin vaiheisiin ja työstetään muutenkin vaiheita hieman limittäin ja lomittain.

Iteraatiolla tarkoitetaan yksinkertaisesti sitä, että pelin kehityksessä suunnitellaan, testataan ja arvioidaan tuloksia uudelleen ja uudelleen parantaen pelin kulkua ja ominaisuuksia, kunnes ne täyttävät kriteerit. Iteraatio on erittäin tärkeää pelaajakeskeisen prosessin kannalta. (Fullerton 2008, 14.) Projektin suunnittelussa painotettiin koko ajan palvelumuotoilulle ominaista iteratiivisuutta. Myös tekemistä pidettiin tärkeänä. Huumorin keinoin toteutettu peli toimisi tässä tapauksessa raskaiden aiheiden käsittelykeinona parhaiten, kun kohderyhmänä on kuitenkin nuoret. Tämä oli myös toimeksiantajan toiveena.

Olio-ohjelmoinnilla tarkoitetaan ohjelmointitapaa, jossa muodostetaan luokkia, joiden pohjalta muodostetaan instansseja, olioita kyseisistä luokista; luokkaan on määritelty kullakin oliolla olevat ominaisuudet, jotka löytyvät jokaisesta yksittäisestä oliosta. Olio-ohjelmoinnissa on kyse sellaisen järjestelmän mallintamisesta, joka koostuu olioista, joista jokainen edustaa jotakin järjestelmän osaa. Olioihin voi sisällyttää dataa ja funktioita tai metodeja. Olio tarjoaa käyttöliittymän muille koodin osille, joiden on tarvetta käyttää sitä, mutta sisältää sen oman, tämänhetkisen tilan, joka on oliolle yksityinen. Näin muiden koodin osien ei tarvitse huolehtia siitä, mitä olion sisällä tapahtuu milläkin hetkellä. (Mozilla 2023a.)



Kuva 1: Pelin aloitusnäky

JavaScriptissa suurin osa asioista on olioita. Olioita ovat esimerkiksi sisäänrakennetut JavaScript-ominaisuudet kuten taulukot ja ohjelmointirajapinnat, jotka on rakennettu JavaScriptin päälle suoraan käytettäväksi. JavaScriptissa on mahdollista luoda omia olioita, jotka sisältävät olion tarvitsemia funktioita tai muuttujia käytännöllisissä paketeissa, jotka toimivat datasäiliöinä. (Mozilla 2023b.)

3.4 Projektin tiedostot ja niiden luovutus asiakkaalle

Projektin tiedostot koostuvat pääosin JavaScript-tiedostosta, HTML-tiedostosta ja CSS-tiedostosta. Pelin logiikka ja suurilta osin ulkoasu on kirjoitettu JavaScript-tiedostoon ja siihen liitettyihin kuviin, jotka sijaitsevat peliprojektin kansiossa. Nettisivu, jolle peli on asetettu, on tehty HTML-kielellä. Tämä nettisivu on pääasiassa kuori pelille, mutta siellä on sidosryhmien logoja ja linkkejä näiden verkkosivuille, sekä tietoa pelin tekijöistä. HTML-koodi sisältää myös Canvas-elementin, jolle piirretään tekstiä ja kuvia JavaScript-koodin avulla.

Ohjelman koodi oli kehitysvaiheessa paikallinen kansio tietokoneella, josta se ladattiin GitHub-repositorioon. Siellä koko kehitystiimi pääsi siihen käsiksi. Kansio sisältää kaiken peliin liittyvän ja sen sisältämän koodin voi ajaa tietokoneella suoraan verkkoselainta käyttäen. Peli

luovutettiin asiakkaalle verkkoon ladatun verkkosivun muodossa sekä ladattavana tiedostona. Pelin laajentaminen jatkossa ei kuulu opiskelijatiimille, mutta mikäli myöhemmin ilmenee ohjelmointivirheitä tai sisällöllisiä virheitä, voi pelin omistaja olla yhteydessä pelin tekijöihin.

3.5 Aikataulu

Projektin aikataulu muuttui projektin edetessä. Tarinan syntyminen meni vuoden 2023 puolelle samoin kuin hahmojen luominen, tosin näistä molemmista oli luonnoksia jo loppuvuodesta 2022. Ensimmäinen prototyyppi luotiin marraskuussa 2022 ja varsinaisen pelin ohjelmointi aloitettiin tammikuussa 2023, kun tarina ja hahmokonseptit alkoivat olla paremmin selvillä. Testattava prototyyppi valmistui ja julkaistiin testikäyttöön 20.6.2023.

Taulukko 1: Alustava aikataulu peliprojektille (Toteutunut aikataulu löytyy tämän raportin kappaleesta 8.4).

Elokuu 2022	Syyskuu 2022	Lokakuu 2022	Marraskuu 2022	Joulukuu 2022	Tammikuu 2023	Helmikuu 2023
projektin aloitus	ideointia ja suunnittelua: mm. workshop nuorille, benchmarking yms. Aiheanalyysi		opparisuunnitelma +seminaari	Pelin kehittämistä ja testausta	Pelin kehittämistä ja testausta	Valmis peli toimeksiantajalle Opinnäytetyön julkaisu
Pelin suunnittelua ja kehittämistä: visuaalisuus, teknisyyt (ohjelmointi)						

Työtuntien laskemista yritettiin projektin alussa, jotta olisi saatu jonkinlainen kuva siitä kuinka paljon työtunteja pelin kehittämisen eri osa-alueisiin kuluu. Myös opinnäytetyön ohjaukseen liittyvissä asioissa oli projektin alussa hieman sekaannusta johtuen monialaisuudesta, mutta loppujen lopuksi kaikki järjestyi hyvin ja työn ohjaajana toimi tietojenkäsittelyn lehtori. Restonomi-opiskelija pääsi oman koulutusalsansa opinnäytetyön tekemistä ohjaavaan työtilaan, joka auttoi myös paljon tämän työn sekä raportin rakentumisessa, koska työtilan sisältö oli erinomainen ja kattava.

4 Taustatutkimus

Esitutkimuksessa selvitetään tuotteen tai palvelun ominaisuuksia muihin vastaavanlaisiin tuotteisiin ja palveluihin vertailemalla (Raatikainen 2008, 61, 62). Tämän työn tavoitteena on siis luoda uusi digitaalinen palvelu, joka on myös samalla tuote. Palvelumuotoilun ensimmäisessä vaiheessa on tarkoitus löytää käyttäjän mielenkiinnon kohteita ja tutkia hänen käyttäytymistään, motiiveita, sosiaalisuutta ja unelmia ympäristöissä, joissa hän elää ja toimii (Koivisto, Säynäjäkangas, Forsberg, 2019, 44). Tehty empatiakartta auttoi hahmottamaan loppukäyttäjän tunteita ja ajatuksia.

4.1 Benchmarking

Tutkimuspohjaiset pelit moninkertaistavat terveystietoisuusmenetelmien vaikutuksen ja tehokkuuden, mikä on lupaavaa esimerkiksi tulevaisuuden mahdollisuuksille pelillisessä diagnostiikassa ja terapiassa (Psyongames 2023). Tuotetta varten benchmarkattiin oppimispelejä: suomalaisen Psyon Gamesin Full ADHD - täyttää elämää, YLE:n Trollibunkkeri, Universeandmoren Crack the Circuit, Arjenturvallisuuden Stop-peli sisäisestä turvallisuudesta, Ruokatuokan Ravitsemuspeli ja Nuortenlinkin peli. Lisäksi joukkoon valikoitui Steam-pelialustalta löytyvä Reigns-korttipeli.



Kuvio 3: Havainnot ja johtopäätökset

Ruokatutkan ravitsemuspeli sekä Nuortenlinkin peli olivat sisällöltään asiallisia sekä kattavia, helposti pelattavia, hauskoja ja simppeleitä tarinallisia valintapelejä. Suomalaisen Psyon Gamesin Full ADHD - täyttä elämää on hyvä esimerkki upeasta visuaalisuudesta ja pelattavuudesta. Universeandmoren Crack the Circuitissa äänimaailma oli toteutettu erinomaisesti. YLE:n Trollibunkkerissa oli selkeä ohjeistus pelin käyttöön sekä myös mielenkiintoinen ja jännitystä herättävä äänimaailma.

Stop-peli arjen turvallisuudesta osoittautui hyvin hyödylliseksi benchmarking-kohteeksi. Peli oli nimenomaan suunnattu yläkouluikäisille, kuten tämänkin projektin peli. Peli oli myös toteutettu korttipelityyppisenä valintapelinä ja vaikutti visuaalisesti hienolta ja siihen oli saatu sisältö liitettyä yhteen erittäin kiinnostavalla tavalla. Valitettavasti peli oli poistunut sivustolta loppuvuoteen 2022 mennessä, eikä sitä ole enää verkossa saatavilla.

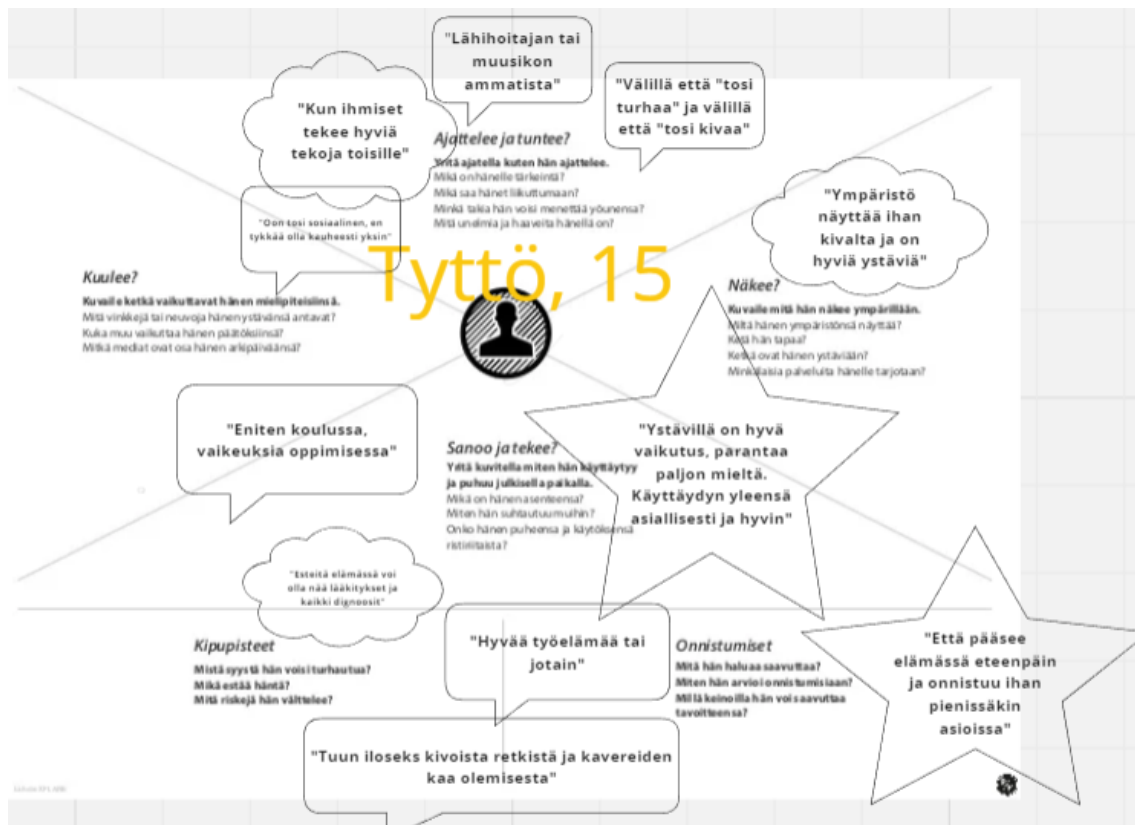


Kuva 2: Arjen turvallisuuden STOP-peli.

Fantasiakesälomapelissä ei ole mahdollista valita vaikeustasoa, vaikka yleisesti ottamalla huomioon pelisuunnittelussa vaikeustason valinta, voidaan vaikuttaa pelaajan saamaan pelikokemukseen paljonkin (Pelikasvattajan käsikirja 2023).

4.2 Empatiakartta

Empatiakartta on työkalu parempaan asiakasymmärrykseen, sillä hahmotetaan asiakkaiden tarpeita mm. kysymällä ja pohtimalla minkälaista elämää tämä elää (Maula, Maula 2019, 53). Käyttäjien emotionaalinen ymmärtäminen empatiakartan avulla on yhtä tärkeä osa tuotekehitysprosessia kuin tekninen käyttäjäkokemusvirran tarkastelu (Storyboard That 2023).



Kuvio 4: Empatiakartta.

Eettisen tutkimustyön suuntaviivojen mukaisesti sähköpostitse haastatellun 15-vuotiaan tytön vastaukset julkaistaan tässä raportissa nimettömänä ja hän on vapaaehtoisesti omalla suostumuksellaan osallistunut pelin taustatutkimukseen antamalla arvokasta tietoa pelintekijöille siitä, mitä hän kokee, tuntee ja ajattelee.

Empatiakartassa merkittäviksi nousseita asioita olivat selvästi ystävät, jotka olivat kohderyhmän edustajalle tärkeitä. Jos ohitamme nämä tärkeät havainnot, jotka selvästi nousevat esiin käyttämiemme palvelumuotoilun menetelmien ansiosta, emme voi sanoa suunnittelevamme asiakaslähtöisesti. Asiakaslähtöinen suunnittelu toimi projektissa hyvin. Tietyt teemat nousivat esille sekä empatiakartassa sekä syksyllä 2022 käydyn workshopin aikana. Eräs selkeä havainto on ystävien merkitys nuorille. Tosin kohdehenkilö mainitsee olevansa hyvin sosiaalinen ja viihtyy seurassa. Ei ole tietoa, tarkoittaako hän sosiaalisuudella sosiaalista mediaa vai jotakin muuta yhdessä olemista. Oma imago on yllättävän tärkeä nuorille ja sen rakentuminen 14-18 vuoden ikäisenä voi olla valtava osa ihmisen persoonallisuutta. Pidetyssä workshopissa ja empatiakartassa eräiden tyttöjen mielestä kiinnostavaa olivat mm. meikit ja hahmon luominen; asujen valinta ja tyyli. Tulevaisuudensuunnitelmat ja ammatinvalintaan liittyvät kysymykset voivat myös herätellä nuoria pohtimaan valintojaan vaikkapa rahaan liittyen.

5 Määrittely

Asiakastarpeiden määrittelyyn on olemassa eri menetelmiä, näitä ovat esimerkiksi palvelupolut (Koivisto ym. 2019, 46). Palvelumuotoiluprosessin 1. palvelupolku on kuvitteellinen. Tässä projektissa todenperäisempää ja palvelumuotoilun prosessissa mahdollisesti myöhemässä vaiheessa tehtävää 2. palvelupolkua ei toteutettu. Palvelutuotekehittämisen näkökulmasta oppimispelin määrittely taustoitettiin käyttäjakeskeisen suunnittelun opintojaksoon kuuluvassa tutkimuksessa jo keväällä 2022, kun tietoa kerättiin kyselyllä Munuais- ja maksaliiton työntekijöiltä, liiton jäseniltä/digitaalisten palvelujen käyttäjiltä sekä sidosryhmiltä kuten sairaanhoitajilta. Silloin liitolle tarjoutui mahdollisuus palvella nuoria erityisemmin ja kohdistetummin. Kuitenkin itse pelituotteen muotoiluun tarvittiin vielä lisätyökaluja. Myös sisällölliset tavoitteet ominaisuuksien lisäksi määriteltiin tarkemmin syksyllä 2022.



Kuvio 5: Toimeksiantajan ja yhteistyökumppaneiden sanoja pelin sisällöstä.

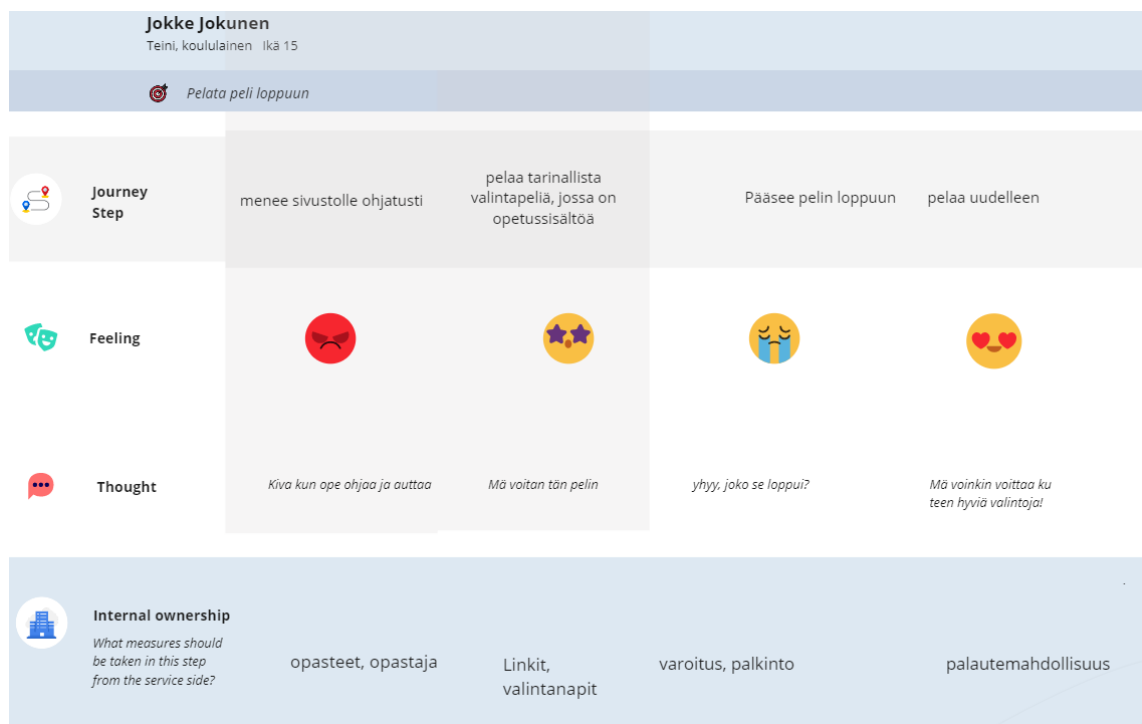
Sisällöllisten tavoitteiden asettaminen oli pelin lopputuloksen kannalta merkittävää. Tavoitteiden asettaminen antoi hyvän pohjan projektityöskentelylle. Projektin joustavan luonteen vuoksi tarkkarajaisia ehdottomasti saavutettavia tavoitteita ei kuitenkaan asetettu.

5.1 Oppimispeli

”Nykyisenkaltaiset oppimispelit ovat kehittyneet 1980-luvulta lähtien; ensin drilli- tyyppisinä yritä- erehdy- opi- peleinä ja multimediatietokoneiden aikakauden myötä audiovisuaalisina elämyksinä” (Toim. Krokfors ym. 2014, Ängeslevä, 120). Koska kyseessä on palvelutuote, voitiin siitä tehdä kuviteltu palvelupolku, jossa nuori käyttäjä ottaa käyttöönsä pelin. Mistä peli löytyy? Löytyykö se helposti? Todennäköisesti pelaaja ei eksy vahingossa pelaamaan, vaan

hänet ohjataan pelaamaan sitä liiton sivustolta. Peliä saatetaan käyttää myös opetuksellisessa yhteydessä kouluissa, jolloin opettaja kertoo mistä osoitteesta peli löytyy. Murrosikäistä voi kiukuttaa tämä tai jokin muu asia. Nuorten tunteet muuttuvat, kun he näkevät ja kuulevat kivan sekä kauniin pelin. Tämä onkin hauskaa ja jännittävää oppilaan mielestä, innostuneena hän klikkailee tarinaa sisältöineen eteenpäin. Pisteiden kerääminen ja menettäminen jatkuu, kunnes peli loppuu. Lopussa pelaajalle annetaan palaute ja kerrotaan, miten kävi, tuliko tehtyä hyviä valintoja vai ei. Ihastuneena utelias pelaaja ei malta jättää pelaamista yhteen kertaan vaan pelaa peliä toisen ja kolmannenkin kerran, jolloin tarina ja lopputulos muuttuvat. Tämän toivotaan vahvistavan sisäistä motivaatiota siten että pelaaja kokee vaikuttavansa tapahtumiin pelissä sen lisäksi, että pelaaja löytää sitä enemmän sisältöä pelistä mitä enemmän hän sitä pelaa. Se on hauskaa.

Taulukko 2: Fantasiakesälomapelin Palvelupolku.



”Pelien suunnittelu ja tuotanto on paitsi audiovisuaalista taidetta, myös systeemidynamiikkaa sekä motivaation ja käyttäytymisen psykologiaa. Nykyisten pelien valtavat käyttäjämäärät ja pelaajien käytösdatan reaaliaikainen kerääminen mahdollistavat koeasetelmia, joilla pelien koukuttavuutta ja liikevaihtoa voidaan optimoida tehokkaasti. Käytösdata ei kuitenkaan kerro kaikkea siitä, mikä pelaajia ohjaa ja motivoi. Ulkoisia pelimotivaatioita ovat esimerkiksi palkinto tehtävän suorituksesta. Sisäisiä motivaatioita on esimerkiksi luontainen uteliaisuus. Hyvän pelisuunnittelun ydin on sisäisen motivaation ymmärtäminen ja tukeminen, joka ei kuitenkaan ole yhtä suoraviivaista kuin ulkoinen motivointi.” (Hämäläinen, Takatalo, 2017.)

5.2 Pisteytys

Pelaaja saa valintojen mukaan pisteitä eri luokissa. Pelin tarkoituksena ei ole rankaista pelaajaa valinnoista, vaan tarjoamalla erilaisia vaihtoehtoja opettaa ja opastaa nuorta miettimään valintojensa seurauksia. Pelaaja saa pisteitä valinnoistaan neljässä eri luokassa: terveys, pisteet, raha ja sosiaalisuus. Alustavasti oli puhetta siitä, ettei mikään valinnoista olisi ns. väärä valinta. ”Väärin tekemisessä kuitenkin piilee myös dramaturginen koukku, joka tekee pelaamisesta jännää ja mielenkiintoista (Vaikkola 2022).”



Kuva 3: Pisteet pelin prototyypissä.

Pisteet menevät yleisten pisteiden lisäksi kategorioihin Raha, Terveys ja Sosiaalisuus. Pelissä on mietitty mikä on tärkeää kaikille nuorille kesäloman aikana ja lopussa kolme kategoria nousi esille. Pelissä on mahdollista niin menettää kuin myös saada lisää pisteitä päätösten takia, pelin prototyypissä kesätöistä pelaaja sai varoja, mutta ei viettänyt niin paljoa aikaa sosialisoidessa pelin hahmojen kanssa. Tämä idea karsittiin lopullisesta pelistä, mutta se vaikutti taustalla tarinan ja peliominaisuuksien suunnitteluun ja toteutukseen.

Terveyttä voi menettää kokeilemalla erilaisia päihteitä, toisin sanoen huonojen päätösten aiheuttamana. Terveyspisteitä voi myös saada lisää terveellisten elämäntapojen ja päätösten pohjalta. Terveiden kaltaisen resurssin käyttäminen tarkoittaa yleensä sitä, että on olemassa tapa parantaa terveyttä, vaikka se menetettäisiin osana peliä (Fullerton 2008, 73).

Sosiaalisuus on pistekategoria, joka vaikuttaisi olevan varsinkin yläkouluikäisille tärkein. Tämän takia sosiaalisuuspisteitä on helppo saada, mutta myös helppo menettää pelissä ja osa pelissä tehdyistä päätöksistä tai vuoropuhelusta vaikuttaa vasta myöhemmin pelissä. Vaikutus näkyy vasta pelin lopussa, tämä auttaa pelaajaa ymmärtämään tekemiään päätöksiä ja puhevalintoja useammasta kuin yhdestä näkökulmasta.

Pisteet ovat pelin edetessä kerättäviä ja ainut ohjenuora pelaajalle, että hän on tehnyt oikean päätöksen. Pisteitä tosin saadaan vasta päätöksen jälkeen, näin pelaajalle tulee vasta valinnan tehtyään tieto pisteiden noususta tai laskusta.

Suurin osa päätöksistä pelissä on kuitenkin vihjaavia: jos pelaaja saa eteensä mahdollisuuden kokeilla alkoholia, on oletettavaa, että terveys kärsii. Muutamassa kohdassa pelaajan on käytettävä rahaa ja tämän peli tekee selväksi, mutta pelaajalla on mahdollisuus kieltäytyä. Tällä on ideana opettaa nuoria rahan käytöstä kavereiden kanssa. Huomattavin pistekategoria on

sosiaalisuuspisteet, joihin voi helposti vaikuttaa pelaajan päätöksillä. Ryhmäpaine on otettu huomioon pelissä ja juhlissa ainut henkilö, joka ei juo alkoholia saattaa erottua muista. Myös arkinen keskustelu pelin hahmojen kanssa on usein sosiaalisuuspisteitä muuttava tapahtuma.

Pelissä on neljä tarinalukua. Näiden lisäksi pelissä on introluku, jossa esitellään peliä sekä loppuluku, jossa näkyy pelaajan pelin aikana tekemien valintojen vaikutukset. Loppuluvun jälkeen pelaaja näkee kaikkien pistekategorioiden pisteet, loppuruudussa hänelle annetaan myös mahdollisuus pelata peliä uudestaan.

Tarinaluvuista pelaaja pääsee kahteen kullakin pelikerralla. Peli alkaa introluvulla. Introluvun jälkeen pelaajalta kysytään, meneekö hän Ilta-hahmon lukuun vai Roihu-hahmon lukuun. Kummastakin näistä luvuista pelaaja pääsee joko Mantan tai Laineen lukuun, näiden jälkeen hän pääsee loppulukuun. Kullakin pelikerralla kahteen lukuun neljästä pääsemisellä on saatu peliin pituus kohtuulliseksi ja lisätty pelin uudelleenpelattavuutta, samalla mahdollistaen uusien asioiden oppimista.

6 Pelin ideointi ja suunnittelu

Erilaisiin asiakastarpeisiin vastaaminen edellyttää asiakasymmärrystä. Asiakasymmärryksen lisäksi suunnittelussa käytetään erilaisia ideointimenetelmiä ja rakennetaan prototyyppejä. Ideointiin osallistetaan työpajoissa henkilökuntaa, asiakkaita ja sidosryhmiä. Yhteiskehittämällä saadaan monimuotoisempaa näkemystä ja asiantuntemusta ongelmanratkaisuun (Koivisto ym. 2019, 46). Tuotekehityksessä (eng. Product development) saadaan näin aikaiseksi palveluita, jotka toimivat paremmin (Bergström, Leppänen, 2016, 183).

Palvelumuotoilun toisessa vaiheessa oli tarkoitus ideoida tuotetta. Ideointia toteutettiin opiskelijatiimin kesken, toimeksiantajan kanssa, sekä myös järjestämällä ideointipaja nuorille, jotka kuitenkin ovat tämän palvelutuotteen loppukäyttäjiä. Ideointi kokouksissa yhdessä toimeksiantajan sekä sidosryhmien kanssa toimi hyvin liiton palvelupäällikön fasilitoimana. Fasilitoinnissa on kyse hyvistä ja selkeistä kokouskäytännöistä, joilla erilaisia ihmisiä kehoitetaan sekä kannustetaan osallistumaan keskusteluun ja päätöksentekoon (Sipponen-Damonte M. 2020, 39). Koko projektiorganisaation tapaamisia järjestettiin n. 1,5 kk:n välein. Opiskelijatiimi tapasi useammin. Kokoonpanot ja osallistuminen oli molemmissa tapauksissa vaihtelevaa.

6.1 Ideointi

Yläkouluikäiset kohderyhmänä haastoivat tiimiä pohtimaan, minkälainen peli palvelisi juuri tätä kohderyhmää parhaiten ja varsinkin tämän päivän diginatiivisuus herätti ajatuksia sekä keskustelua. Nuoret ovat hyvin nopeaälyisiä ja saattavat kyllästyä nopeasti. Heille olisi hyvä tarjota visuaalisesti kiinnostava peli, jossa kuitenkin ymmärrettäisiin nuorten elämän tapahtumapaikkoja sekä toimintaympäristöjä. Nuoret elävät vahvasti sosiaalisissa ympäristöissään ja ystävät ovat heille todella tärkeitä. Opiskelijatiimin jäsenillä ja muilla projektiin osallistuvilla oli työyhteyksien lisäksi myös omassa tuttava- harraste- ja perhepiirissä kohderyhmään kuuluvan ikäryhmän edustajia, mikä toi työskentelyyn omakohtaisuutta ja lisäännostusta.



Kuvio 6: Tärkeinä ja hyvinä pidettyjä pelin ominaisuuksia.

Kohderyhmää kuullakseen järjestettiin nuorille kehitysriihi/workshop 7.10.2022 klo 14-15.30 Munuais- ja maksaliiton tiloissa Helsingin Pasilassa. Paikalla oli opiskelijoiden lisäksi 8 nuorta, joista 2 oli poikia ja 6 tyttöjä. Nuorten workshopin tarkoituksena oli nimenomaan kerätä kohderyhmän eli nuorten ajatuksia pelillisyydestä, ei niinkään opettusisällöstä. Pelillisyyttä

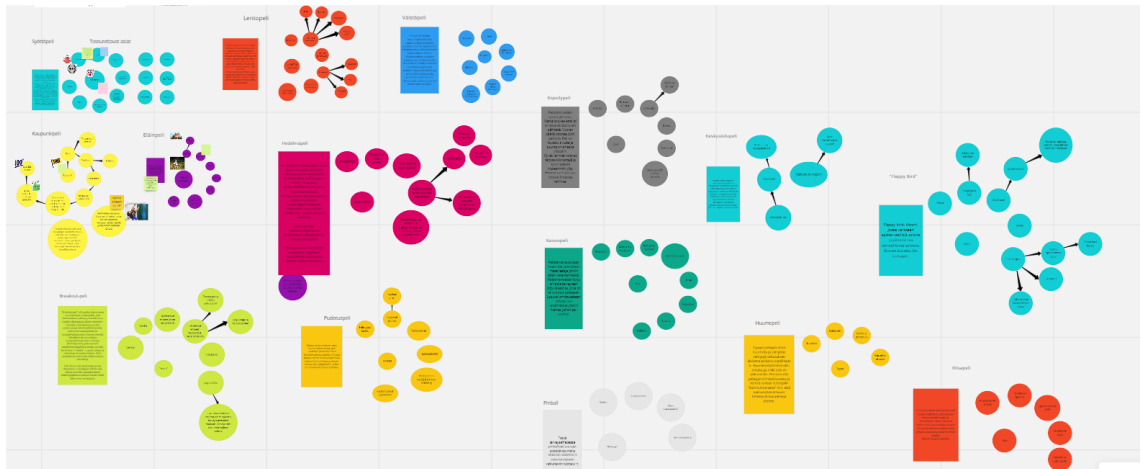
käsitetään tässä yhteydessä pelin pelattavuuden osatekijänä. Workshopin kysymykset löytyvät liitteestä.



Kuvio 7: Huonoina pelin ominaisuuksina mainitut.

Omassa ideoinnissaan opiskelijat käyttivät Miro-alustaa, jolla ideointi onnistui helposti ja vaihtavasti. Ideointivaiheessa oli tavoitteena tuottaa mahdollisimman paljon ideoita. Peli-ideoita olivat tässä vaiheessa mm. Syöttöpeli (hahmolle syötetään asioita), Kaupunkipeli (pelaajalle näkyy kaupunki, jossa valitaan tehtäviä), Breakout-peli (rikotaan pallolla palikoita joista saa pisteitä), Eläinpeli (tarinoita eläinhahmoilla), Lentopeli (ammutaan vihollisia lentävällä hahmolla), Hedelmäpeli, Pudotuspeli (hahmolle tiputetaan ruokaa), Väistöpelit (väistetään esineitä nopeasti liikkuvalla hahmolla) Kiipeilypeli (kiivetään vuorta ylöspäin ja varotaan esteitä), Kaivospeli (kerätään esineitä kaivautumalla maahan), Pinball (flipperipeli), Keskustelupeli (keskustellaan esim. huumeidenkäyttäjien kanssa), Huumeopeli (yritetään selviytyä

tilanteesta huumattuna), Flappy Bird (Flappy Bird -kloonin, väistetään esteitä lentäen), Ritsa-peli (ohjataan hahmoa ampumalla ruokia sitä kohden).



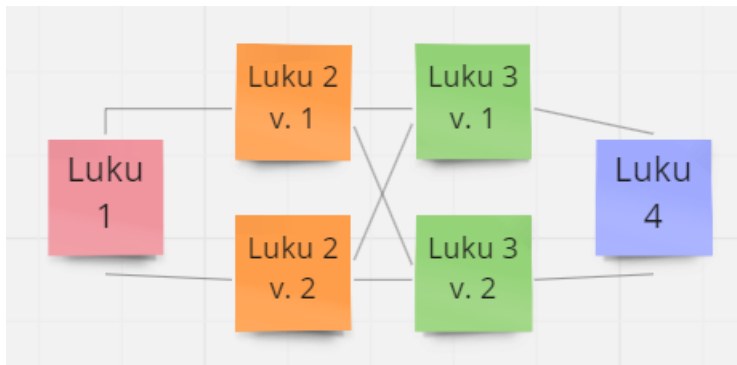
Kuvio 8: Ideointia Mirossa. Peli-ideat ovat omissa kategorioissaan eri väreillä merkattuina.

Opiskelijatiimin jäsenet siisideoivat yhdessä ja erikseen myös itse, mutta myös suunnittelijoilta, toimeksiantajalta sekä sidosryhmiltä kerättiin ajatuksia pelistä. Ideointivaiheen lopussaideoista karsittiin selkeästi huonot ideat pois ja vertailtiin jäljelle jääneitä toteuttamiskelpoisia ideoita. Lopulta päädyttiin Reigns-pelin inspiroimaan korttipeliin, joka toimii visuaalisen novellin tyyliä kuva- ja tekstikerronnan sekä taustamusiikkien avulla. Pelissä yhdistyy fantasiamaailman audiovisuaaliseen tarinankerrontaan perustuva sisältö sekä pelissä esiintyvien eläinavustajien kertomat tietoisuudet, jotka sisältävät nuorisolle merkityksellistä terveystietoa.

6.2 Prototyyppi

Hyvä pelisuunnittelija pitää prosessia idean kehittämisestä ja alkuperäisestä konseptista pelattavaksi sekä tyydyttäväksi pelikokemukseksi tärkeänä. (Fullerton 2008, 10). Konsepti oli pitää peli opettavaisena ja hauskana pelattavana lapsille. Peli sijoittuu kesälomalle fantasiamaailmaan. Ideana on edistää lasten terveystietoisuutta niin maksan kuin myös munuaisten terveydestä. Myös muu terveysopetus muista aiheista kuten päihteiden haitoista on sisällytetty peliin. Hieno juttu on, että tilastojen mukaan raittiiden alaikäisten nuorten osuus on kasvussa, eivätkä esimerkiksi alkoholikokeilut enää automaattisesti kuulu nuoruuteen (Mannerheimin Lastensuojeluliitto 2022).

Pelin ensimmäinen prototyyppi päätettiin tehdä Reigns-pelin tyyppisenä korttipelinä. Peli sijoittuu fantasiamaailmaan, jossa seikkailevat piirroshahmot. Aiheena on: Kuinka hyvin selvisit kesälomasta/kesäloma. Tarina alkaa simppelellä, koetetaan saada pelaaja kiinnostumaan pelistä ja tajuamaan miten peli toimii, jokainen mittari tekee asioita ja vaikuttaa peliin.



Kuvio 9: Pelin suunniteltu eteneminen kaavion kautta Mirossa.

Korttipeliin tulisi terveydellisiä ja hyvinvointiin liittyviä teemoja. Jokaiselle terveydelliselle teemalle on oma hahmonsa ja lukunsa (koodissa chapter). Jokainen kortti ei ole opetuksellinen. Aiheet tuodaan esille tarinassa luontevasti, jotta tarinassa ei ole töyssyjä tai niin sanottuja Big Lipped Alligator -hetkiä (TV Tropes 2023). Kortissa näkyy tapahtuma ja kortin taustassa sekä sen takana miljö, jotka auttavat eläytymään pelitapahtumaan. Taustakuvan ja peliruudun itsensä mitat ovat 1024x768 pikseliä isolta näytöltä katsottuna ja 800x600 pikseliä pieneltä näytöltä katsottuna. Pelin koodi mittaa automaattisesti näytön koon. Tekstiruudussa on tarinan tekstiä. Pelissä ideana on valita asioita. Mittaria ”aikaa jäljellä kesän loppuun” ei määritelty, vaikka tästä oli puhetta suunnitteluvaiheessa, idealee ei löytynyt hyvää toteutustapaa. Pelissä on noin 290 korttia ja yhden pelikerran kesto on n. 10-15 minuuttia. Jokaista korttia ei läpikäydä joka pelikerralla, korttien suuri lukumäärä mahdollistaa tarinan haarautumisen niin, että loppuja ja tarinan kulkuja on useita. Kokemus voi siis olla hyvinkin erilainen eri pelikerroilla. Peliä voi pelata opetuksellisesti vaikkapa koulun oppitunnilla tai nuorisotilan tietokonehuoneessa. Peli on innostava, mielenkiintoinen ja nuorille sopiva. Kortit eivät sisällä animaatioita, mutta niihin voisi myös liittää liikettä. Kuvittamisen kannalta kortit toimivat hyvin, niiden kuvittaminen oli graafikon mielestä helppoa ja selkeää. Kortin kuvien mitat ovat 300x300 pikseliä isolla ruudulla ja pienellä ruudulla 250x250 pikseliä.



Kuva 4: Roihun kappaleen avustaja ja sen kertoma teksti yhdessä tarinan kohdassa.

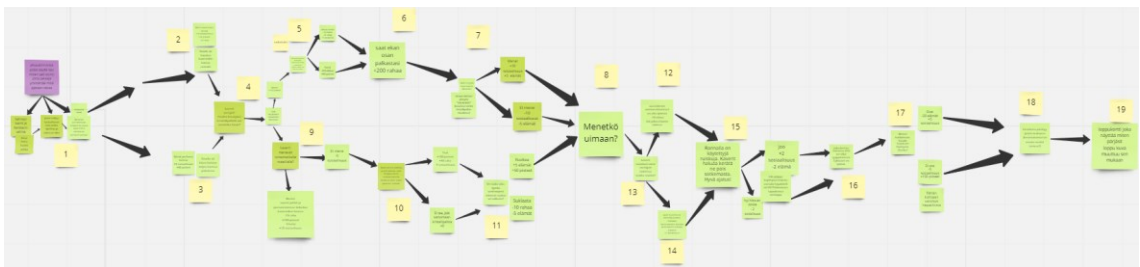
Pelin hahmot ovat piirroshahmoja ja taustat Stable Diffusion -tekoälysovelluksella generoituja. Pelissä ja luvuissa on selkeä alku ja loppu, joiden väliin jää muuttuva tarina itsessään, joka haarautuu eri skenaarioihin. Kortit ovat numeroituja. Peliin liitettiin eläinhahmon, niin sanotun avustajan kertomia tietoiskuja ja pelialustaan linkkejä sidosryhmien verkkosivuille. Pelissä avustaja kertoo terveyteen liittyviä faktoja. Avustaja osoittaa tärkeimmät päätökset ja neuvoo, mutta koska kyseessä on valintapeli, pelaaja päättää itse. Jokaisella pelin kappaleella on oma graafikon piirtämä avustaja. Avustajan kuvan mitat ovat 128x128 pikseliä.

6.3 Prototyypin tarina

Usein terveyspeleissä on paljon tekstiä. Pidettiin tärkeänä, ettei peliin tule ns. infodump-pausta, eli niin paljon informaatiota kerralla, että se on ylikuormittavaa. ”Toisaalta nuorilla on kyky lukea ja omaksua nopeasti tekstiä, joten suurikin määrä kiinnostavana pidettyä tarinallista dialogia ei haittaa, kunhan sen tarkoituksena on saada pelaaja eläytymään peliin sekä kuljettaa pelaajaa eteenpäin” (Vaikkola 2022). Melko pitkätkin tekstit on saatu näyttämään runollisilta sommittelemalla ne kortteihin. Dialogi pelin hahmojen kanssa mahdollistaa myös sen, että valittaessa pelaajalle annetaan mahdollisuus sanoa muutakin kuin kyllä tai ei. Josain tapauksessa selkeä ei-vastaus on ihan hyvä vaihtoehto vaikkapa terveyden näkökulmasta. Pelissä on ainoastaan suljettuja kysymyksiä johtuen pelin rakenteesta: jokaisessa kortissa on kortista riippuen yhdestä kolmeen vastausvaihtoehtoa. Suljetut kysymykset eivät salli vastajan antaa omaa vastausta, vaan vastaus on valittava ennalta määrätystä vaihtoehdoista, kuten vaikkapa haluaako joku makaronilaatikkoo vai lihapullia sen sijaan, että kysyttäisiin ”Mitä haluaisit ruoaksi?” (SurveyMonkey 2023).

Tulevaisuudentutkija Sohail Inayatullahin mukaan pelaaminen antaa puitteet tarkastella näkökulmia, jotka muuten jäisivät helposti marginaaliin ja mahdollistaa uusien, vielä nupullaan olevien ajatusten syntymisen. Myös futuristi ja suunnittelija Stuart Candy näkee pelit työkaluna rohkaista ihmisiä omaksumaankin tutkiva ajattelutapa, jolloin ei vain yritetä ajatella toisin, vaan tarjotaan konkreettista materiaalia mielikuvituksen tueksi. (Poussa, Ylikoski, 2022.)

Peli suunniteltiin niin, että peliä pelatessa pelaajan on käytettävä mielikuvitustaan tarinan elävöittämiseksi, näin oppimateriaali pelistä jää mahdollisesti mieleen paremmin, kun opetusisältö kiinnittyy pelaajan mielikuviin pelistä. Usein sanotaan, että aivot oppivat visuaalisesta ärsykkeestä paremmin kuin pelkästä lukemisesta.



Kuvio 10: 1. Prototyypin tarina Mirossa

Kesä tarkasteluvälinä on nuorille ajanjakso, jolloin he eivät ole koulussa ja viettävät aikaa enemmän itsekseen sekä ystäviensä kanssa. He ovat silloin vapaampia tekemään omia valintojaan. Valintapelissä toteutuu hyvin asiakkaan lähtökohtainen toive pelistä ja siitä, kuinka saada nuori pohtimaan omien valintojensa seurauksia. Pelissä olisi voinut olla tekstiosuuksien lisäksi myös puhuttuja osuuksia, jos tämän toteuttaminen olisi ollut realistista. Kaikki hahmot eivät esiinny pelissä kerralla. Kesäloma on kuitenkin melko lyhyt, eikä siinä ajassa ehdi tekemään kaikkea.

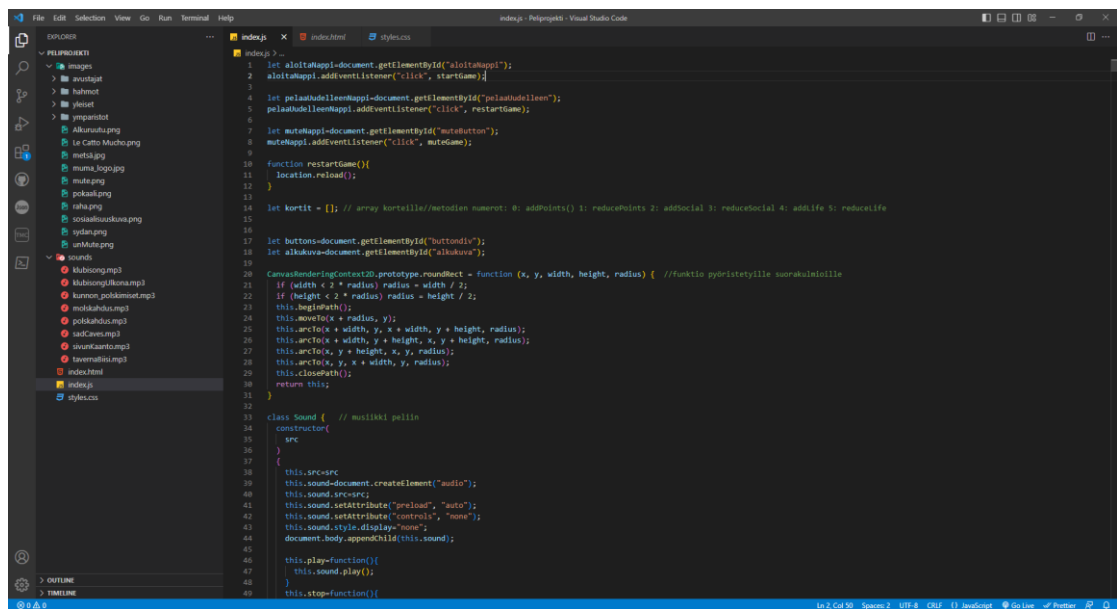
6.4 Prototyypin ohjelmointi

Ohjelmointi pyrittiin tekemään niin, että koodi olisi mahdollisimman modulaarista, jotta sen muokkaaminen ja laajentaminen olisi mahdollisimman helppoa. Funktiot ja metodit, joilla vaikutetaan pelin toimintaan eli pelilogiikkaan, pyritään pitämään tarkkarajaisina ja koodissa hillityn kokoisina, jotta uusien pelielementtien toteuttaminen olisi sujuvaa. Funktiot myös nimetään mahdollisimman selkeästi, jotta peliä olisi helppo lukea koodatessa. Tämä myös mahdollistaa mahdollisen pelin laajentamisen ja saman pelipohjan käytön tulevaisuuden projekteja varten. Yhden kortin ohjelmoinnissa kestää n. 3-5 minuuttia. Tämä ei ole pelin ohjelmointijälle kohtuuton taakka ajallisesti. Pelin koodin yleinen rakenne vakiintui prototyyppivaiheen

lopussa, sen rakenne pysyi projektin loppuun asti pitkälti samanlaisena. Koodia kuitenkin pyrittiin siistimään ja parantelemaan koko projektin ajan.

7 Ohjelman logiikka

Peli on rakennettu olio-ohjelmoinnin avulla. Olio-ohjelmointiin päädyttiin, sillä se mahdollistaa samojen elementtien kätevän toistamisen niin, että elementeillä on eri sisältö. Tämä tarkoittaa tähän projektiin liittyen käytännössä sitä, että pelin sisällön koostuessa virtuaalisista korteista, näitä kortteja on helppo tehdä paljon, mutta niin, että jokaisella on eri sisältö kuvineen ja pistearvoineen. Varsinainen peli on kirjoitettu koodissa Peli-nimisen funktion sisälle. Tätä funktiota kutsutaan heti, kun pelin aloitusnappia on painettu. Kyseisen funktion suoritus myös lopetetaan, kun peliä halutaan pelata uudestaan, tällöin funktiota myös kutsutaan uudestaan. Funktion Peli sisällä on muita funktioita, joita kutsutaan peliä pelattaessa ruudulla näkyvien korttien alapuolella olevia nappeja painamalla, joiden teksti muuttuu pelin tarinan mukaisesti. Myös korttien kuva ja teksti vaihtuvat tarinan mukaisesti, sekä korttien takana näkyvä taustakuva. Käytännössä koko koodi koostuu ensin muuttujien alustamisesta, sitten luokkien määrittelystä, funktioiden määrittelystä, olioiden muodostamisesta luokkien perusteella ja tämän jälkeen funktioiden kutsumisesta olioiden kautta.



```

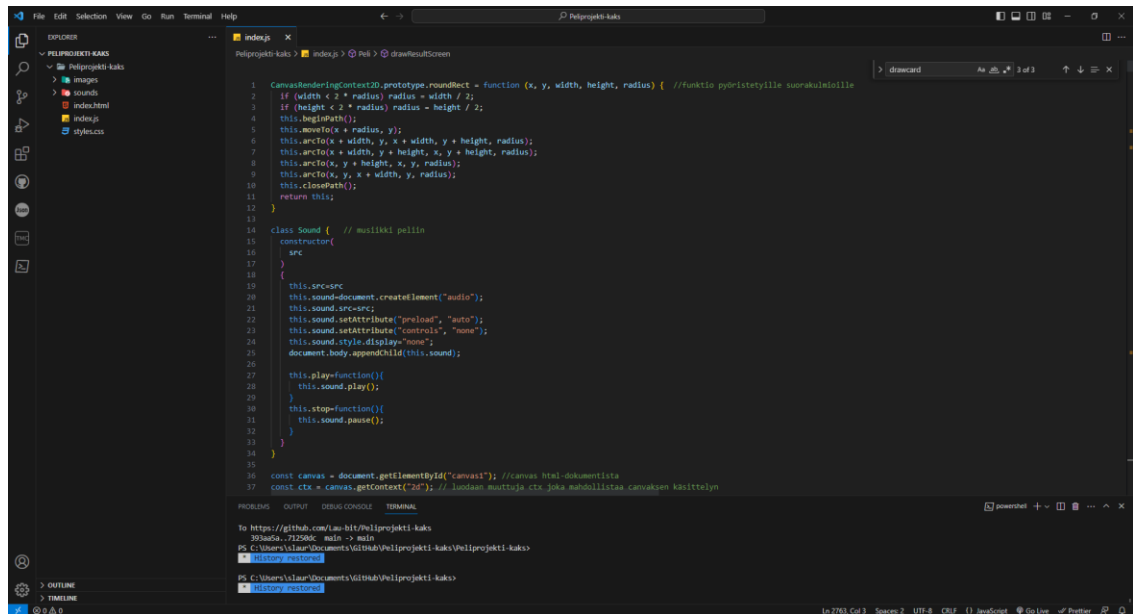
index.js
1 let aloitaAppi=document.getElementById("aloitaAppi");
2 aloitaAppi.addEventListener("click", startGame);
3
4 let pelaakudalleenAppi=document.getElementById("pelaakudalleen");
5 pelaakudalleenAppi.addEventListener("click", restartGame);
6
7 let muteAppi=document.getElementById("mutebutton");
8 muteAppi.addEventListener("click", muteGame);
9
10
11 function restartGame(){
12     location.reload();
13 }
14
15 let kortit = []; // array kortteille/metodien numerot: 0: addPoints() 1: reducePoints 2: addSocial 3: reduceSocial 4: addLife 5: reduceLife
16
17 let buttons=document.getElementById("buttonid");
18 let alkukuva=document.getElementById("alkukuva");
19
20 CanvasRenderingContext2D.prototype.roundRect = function (x, y, width, height, radius) { //Funktio pyöristetyille suorakulmioille
21     if (width < 2 * radius) radius = width / 2;
22     if (height < 2 * radius) radius = height / 2;
23     this.beginPath();
24     this.moveTo(x + radius, y);
25     this.arcTo(x + width, y, x + width, y + height, radius);
26     this.arcTo(x + width, y + height, x, y + height, radius);
27     this.arcTo(x, y + height, x, y, radius);
28     this.arcTo(x, y, x + width, y, radius);
29     this.closePath();
30     return this;
31 }
32
33 class Sound { // musiikki peliin
34     constructor(
35         src
36     ) {
37         this.sound=document.createElement("audio");
38         this.sound.src=src;
39         this.sound.setAttribute("preload", "auto");
40         this.sound.setAttribute("controls", "none");
41         this.sound.style.display="none";
42         document.body.appendChild(this.sound);
43     }
44     this.play=function(){
45         this.sound.play();
46     }
47     this.stop=function(){
48         this.sound.pause();
49     }
50 }

```

Kuva 5: Peliprototyypin tiedostorakenne ja index.js-tiedoston koodia Visual Studio Codessa 22.4.2023

Ohjelmoinnin edetessä koodin rakenne muuttui melko paljon. Kun varsinainen olioiden ja funktioiden välinen toiminta oli kehitetty, merkittäviä muutoksia perusrakenteeseen ei enää tapahtunut, eikä sitä olisi ollut mahdollista toteuttaa käytännössä. Koodi on rakenteeltaan

melko modulaarista, pelin sisältöä ja ominaisuuksia voi hallita luokkia, niiden olioita ja funktioita lisäämällä, poistamalla ja muokkaamalla. Olio-ohjelmointi sopii tämän projektin tuomien kokemusten mukaan palvelumuotoilua hyödyntävään ohjelmointiprojektiin hyvin, sillä sisältöä on helppo muokata asiakkaan tarpeiden mukaiseksi, kunhan ohjelman peruslogiikka on kunnossa. Tämä todettiin toimivaksi käytännön tasolla projektin edetessä.



```

1 CanvasRenderingContext2D.prototype.roundRect = function (x, y, width, height, radius) { //funktio pyöristetyille suorakulmioille
2   if (width < 2 * radius) radius = width / 2;
3   if (height < 2 * radius) radius = height / 2;
4   this.beginPath();
5   this.moveTo(x + radius, y);
6   this.arcTo(x + width, y, x + width, y + height, radius);
7   this.arcTo(x + width, y + height, x, y + height, radius);
8   this.arcTo(x, y + height, x, y, radius);
9   this.arcTo(x, y, x + width, y, radius);
10  this.closePath();
11  return this;
12 }
13
14 class Sound { // musiikki peliin
15   constructor(
16     src
17   ) {
18     this.src=src
19     this.sound=document.createElement("audio");
20     this.sound.src=src;
21     this.sound.setAttribute("preload", "auto");
22     this.sound.setAttribute("controls", "none");
23     this.sound.style.display="none";
24     document.body.appendChild(this.sound);
25     this.play=function(){
26       this.sound.play();
27     }
28     this.stop=function(){
29       this.sound.pause();
30     }
31   }
32 }
33
34 const canvas = document.getElementById("canvas"); //canvas html-dokumenta
35 const ctx = canvas.getContext("2d"); // luodaan muuttuja ctx joka mahdollistaa canvasin käsittelyn

```

Kuva 6: Peliprojektin rakenne ja koodia 6.9.2023. Projektin edetessä koodi muuttui paikoin merkittävästi, mutta kansiorakenne pelin tiedostoille pysyi samana sen muodostuttua.

Peruslogiikan muodostuttua projektin rakentamisessa ohjelmoinnin osalta ei ollut merkittäviä ongelmia. Teknisiä haasteita ja bugeja koodissa ilmeni kuitenkin koko ohjelmointiprosessin ajan. Bugien ja muiden ohjelmoinnissa tapahtuneiden virheiden korjaamiseen kului suuri osa ohjelmointiin liittyvästä kokonaisajasta. Uusien toimintojen rakentaminen itsessään oli melko nopeaa, näistä paljastui myöhemmissä vaiheissa uusia ongelmia, jotka vaativat korjaamista.

7.1 Olio-ohjelmoinnin toteuttaminen pelissä

Peli on rakennettu olio-ohjelmoinnin menetelmillä, joita JavaScript tukee.

Projektin koodissa kaikilla luokan olioilla on samat ominaisuudet samassa järjestyksessä, mutta eri arvot kyseisille ominaisuuksille. Ominaisuuksilla tarkoitetaan tässä tapauksessa korttiolion järjestysnumeroa, kortin sisällä olevaa tekstiä, eri nappeja klikattaessa aktivoituvia funktioita ja arvoja, joilla näitä kutsutaan, klikattavien nappien tekstisisältöä ja tietoa siitä, mikä on seuraavan pelissä esiintyvän kortin järjestysnumero mitäkin nappia klikkaamalla. Kullekin kortille kuuluvaa kuvaa ei ole määritelty itse korttiolion, mutta ruudulle

piirrettävän kuvan valinta ohjelmassa on riippuvainen kortin järjestysnumerosta, jota ilmaistaan koodissa muuttujan index lukuarvolla. Luokille ei ole määritelty erikseen luokkametodeja tässä pelissä, metodit on toteutettu käytännössä niin, että luokkien ulkopuolelle kirjoitettuja funktioita kutsutaan korttiluokista muodostettuihin olioihin viittaamalla. Metodilla tarkoitetaan tässä yhteydessä luokan sisälle kirjoitettuja funktioita: metodien käyttämisen sijaan kukin korttiolio sisältää viittauksen taulukkoon, joka sisältää viittauksen tarvittaviin funktioihin, jotka ovat luokkamäärittelyjen ulkopuolella.

Luokilla on saatu korttien sisällön kirjoittaminen nopeaksi ja loogiseksi ja sillä varmistetaan, että jokaisesta kortista löytyy kaikki tarvittavat ominaisuudet oikeassa järjestyksessä. Luokkia käyttämällä on myös jaettu erilaiset kortit omiin osioihinsa koodissa, kullakin on vain kyseiselle korttityypille tarpeelliset ominaisuudet. Luokkien käyttäminen helpotti koodin kirjoittamista myös siinä mielessä, että Visual Studio Codessa näytetään kirjoittaessa ehdotuksena luokkaan kuuluvat ominaisuudet, jolloin näitä ja näiden järjestystä ei tarvitse muistaa ulkoa.

7.2 Korttiluokat

Pelin kortit ovat korttiluokkien olioita. Korttiluokkia, joista kortit muodostuvat, on erilaisia käyttötarkoituksen mukaan. Useita korttiluokkia tehtiin sen takia, että tarinankerronta ei rajoittuisi korttien teknisiin rajoituksiin, kuten korttiin mahtuvan tekstin pituuteen tai valintavaihtoehtojen vähyyteen. Pelkästään yhdenlaisten korttien tuomat rajoitukset tulivat ilmi pelin tarinaa kirjoitettaessa niin, että tarinaan kuuluvat pitkät, kuvailevat tekstit eivät mahtuneet Kortti-luokan kortteihin. Kortti-niminen luokka oli teknisen toteutuksen alkuvaiheessa ainoa korttiluokka.

```
class Kortti {
    constructor( //constructor arvojen alustam
        id,
        tarinaTeksti,
        teksti,
        napit,
        kylla,
        ei,
        seuraavaKylla,
        seuraavaEi,
        valinta,
        seuraavaValinta
    ) {
        this.id = id;
        this.tarinaTeksti=tarinaTeksti
        this.teksti = teksti;
        this.napit=napit;
        this.kylla = kylla;
        this.ei = ei;
        this.seuraavaKylla = seuraavaKylla;
        this.seuraavaEi = seuraavaEi;
        this.valinta = valinta;
        this.seuraavaValinta = seuraavaValinta;
    }
}
```

Kuva 7: Korttiluokan Kortti toteutus pelissä.

Luokasta Kortti muodostetuille olioille annetaan muodostinosassa määritellyt ominaisuudet, kun oliot muodostetaan. Luokasta muodostetut instanssit saavat kukin arvonsa this-sanalla määritellyistä kohdista, jotka saavat arvonsa itse olioinstanssia muodostettaessa. Näin saadaan systemaattisesti muodostettua lukuisia samanlaisia kortteja, joilla on kaikilla eri sisältö ja vaikutukset pelin toimintaan.


```

class Dialogi {
    constructor( //constructor arvojen alustamiselle new Kortti() mu
        id,
        tarinaTeksti,
        napit,
        kylla,
        ei,
        seuraavaKylla,
        seuraavaEi,
        valinta,
        seuraavaValinta
    ) {
        this.id = id;
        this.tarinaTeksti=tarinaTeksti
        this.napit=napit;
        this.kylla = kylla;
        this.ei = ei;
        this.seuraavaKylla = seuraavaKylla;
        this.seuraavaEi = seuraavaEi;
        this.valinta = valinta;
        this.seuraavaValinta = seuraavaValinta; // valinnainen kolmas
    }
}

```

Kuva 8: Korttiluokka Dialogi, käytetyin korttiluokka pelissä. Muut korttiluokat ovat hyvin samanlaisia ja ovat käytännössä variaatioita tästä korttiluokasta.

Kortti on korttiluokka, jossa pelaaja näkee tekstiä sen sisältämän kuvan ylä- ja alapuolella. Tämä soveltuu tilanteisiin, joissa ylemmässä tekstissä lukee edellisen kortin vaikutukset tai dialogia ja alemmassa kysymys siihen liittyen, mitä pelaaja seuraavaksi valitsee. tKortti on korttiluokka, joka on tehty sellaiseksi, että siihen mahtuu pitkä teksti. Tekstiä on kortin keskellä olevan kuvan ylä- ja alapuolella. Kortin pääasiallisena tarkoituksena on tarinankerronta. Nimi tKortti on lyhenne sanasta tarinakortti. vKortti on korttiluokka, joka on tehty sitä varten, että pelissä olevat vaikutukset vaikuttavat siihen, mitä kortista tapahtuu. Käytännön toteutuksena pelissä tämä näkyy niin, että Iltan osiossa tarinaa pelaajan hahmon nauttima alkoholin määrä vaikuttaa pelin tapahtumiin. Alkoholin määrä tallennetaan muuttujaan attitude, jonka arvon vKortti-luokan olio sitten lukee ja näyttää eri tekstin ruudulla, sekä eri vaikutukset pelin logiikkaan riippuen attitudeen arvosta. Tätä muuttujaa voisi käyttää muihin vastaviin tarkoituksiin, mutta pelin toteutuksessa ei löydetty muita sopivia tarkoituksia sille ennen pelin loppujaksoa. vKorttia käytetään pelin

loppujaksossa, jossa pelaaja näkee valintojensa vaikutukset. vKortti-luokkaa muokattiin tätä varten sellaiseksi, että jokaisessa pelin jaksossa määritellään muuttujan avulla se, mikä teksti kyseisessä kortissa lukee loppujakson aikana.

```
endingChapterOn=0;
endingListener=0;
iltaListener=0;
roihuListener=0;
laineListener=0;
mantaListener=0;
pisteet=0;
rahat=20;
elkut=100;
sosiaalisuus=50;

iltaAttitude=0;
roihuAttitude=0;
laineAttitude=0;
mantaAttitude=0;
```

Kuva 9: Loppujakson muuttujia.

Loppujakson muuttuja `iltaListener` ja vastaavat `Listener`-päätteiset muuttujat kuvassa 9 määrittelevät, mitkä loppujakson kortit tulevat näkyviin. `iltaAttitude` ja muut `Attitude`-päätteiset muuttujat kuvassa 9 määrittelevät, tuleeeko kyseisessä kortissa ”hyvän lopun” vaiko ”huonon lopun” teksti näkyville. Tämän muuttujan arvolla 0 kortissa näkyy pelaajalle hyvä loppu, arvolla 1 näkyy huono loppu. Muuttujan arvoa muutetaan kussakin jaksossa oleviin korttioloihin asetetulla funktiolla. Arvo on oletuksena 0, joten jos tätä ei erikseen muuteta, pelaaja saa hyvän lopun.

Dialogi-korttiluokan tarkoitus on mahdollistaa monipuoliset dialogit pelaajan ja pelimaailman hahmojen välille. Tämä korttiluokka on tehty niin, että pelissä näkyvät kortit ovat leveitä ja niissä on tilaa tekstile kuvan yläpuolella. Kuvan alapuolelta kortin korkeutta on leikattu niin, että vastausvaihtoehdot mahtuu ruudulle kolme kappaletta. Vastausvaihtoehdot on tehty niin, että niihin mahtuu paljon tekstiä yhdelle riville. Huomiona se, että pitkät vastaukset mahtuvat myös muiden korttiluokkien kortteihin, mutta Dialogi-luokan korteissa niitä mahtuu kolme kahden (tKortti-luokan korttien tapauksessa yhden) sijaan. Projektin edetessä Dialogi-luokka muodostui eniten käytetyksi korttiluokaksi pelin dialogivaihtoehtojen muuttuessa syvällisemmiksi aiempien vaiheiden ideointiin nähden. Korttien ja niiden

vastausvaihtoehtonappien suunnittelussa keskeinen haaste oli kaiken kortin sisällön mahdollistaminen rajallisen kokoiselle pelikortille ja vastausvaihtoehtojen mahdollistaminen kortin alapuolelle.

```
function changeStats(yes) {

  if(changeStatsUsed==0){
    changeStatsUsed=1

    // console.log("changeStatsUsed==0")
  }
  if(gameOn==1){
    // muokkaa pisteitä pelissä
    if (yes === 1) {
      //yesButtonClicked()
      for (let i = 0; i < kortit[index].kylla.length; i++) {
        // for loop joka looppaa kortti-olion vaihtoehdot läpi, mahdollistaa
        functionArr[kortit[index].kylla[i].m](kortit[index].kylla[i].v); // e
      }
    } else if (yes === 2) {
      //noButtonClicked()
      try{
        for (let i = 0; i < kortit[index].ei.length; i++) {
          functionArr[kortit[index].ei[i].m](kortit[index].ei[i].v);
        }
      }
    } catch(error){
      console.log(error.message)
      console.log("EI EI OLEMASSA TÄSSÄ KORTISSA, INDEX:", index)//debug
    }
  } else if (yes === 3) {
    //valintaButtonClicked()
    try{
      for (let i = 0; i < kortit[index].valinta.length; i++) {
        functionArr[kortit[index].valinta[i].m](kortit[index].valinta[i].v);
      }
    }
  } catch(error){
    console.log(error.message);
  }
}
}
```

Kuva 10: Funktio changeStats.

Funktiota changeStats käytetään luokkaolioiden kanssa funktioiden kutsumista varten funktiolistasta. Tämän funktion avulla muokataan pelin pisteiden arvoja. Kuvassa näkyy myös bugien löytymiseen liittyviä console.log-kutsuja.

Korttiluokat toimivat käytännössä niin, että ensin määritellään luokka, johon on kirjoitettu kaikki ominaisuudet, jotka annetaan kullekin luokan oliolle. Sitten luokista muodostetaan oliot, joita laitetaan listaan. Listaa luetaan muuttujan index arvolla, näin valitaan tällä hetkellä pelissä oleva kortti. Seuraava kortti ja pistemuutokset määritellään kyllä, ei ja valinta -nappien avulla, jotka vaikuttavat muuttujan index arvoon riippuen valinnasta. Näin saadaan peli etenemään loogisesti. Riippuen kussakin kortissa saatavilla olevista vaihtoehdoista, pistemuutokset ja seuraava kortti valitaan tämänhetkiseen korttiin kirjoitetuista arvoista Kyllä (funktio, jota kutsutaan, kun painaa kyllä) Ei (sama, mutta painettaessa nappia ei) ja valinta (sama, mutta tätä ei ole kuin joissain korteissa). Seuraava arvo muuttujalle index määritellään olion kohdista seuraavaKyllä, seuraavaValinta ja seuraavaEi. Kyllä, ei tai valinta -nappia painettaessa kutsutaan olioon määritellyistä funktioista valinnanmukaiset arvot, jolloin nähdään vaikutukset pisteisiin käytännössä sekä päästään etenemään pelissä indexin arvon muuttuessa. Tällöin kutsutaan myös animaatiofunktioita ja piirretään seuraava kortti. Klikattaessa nappeja siis kutsutaan funktiota yesButtonClicked, noButtonClicked tai valintaButtonClicked, joissa määritellään muuttujan yes arvo asteikolla 1-3. Näistä funktioista kutsutaan funktiota changeStats(yes), eli kutsutaan tätä funktiota parametrin arvolla yes, jonka sisällä kutsutaan functionArr -taulukosta kyseiseen korttiolioon kirjoitettuja valinnan mukaisia funktioita olioon kirjoitetuilla arvoilla. Listaan functionArr on tallennettu pelin pisteitä muuttavat funktiot, sekä pelin logiikan kannalta oleellisia funktioita.

7.3 Funktio changeStats

Tämän funktion tarkoituksena on muuttaa pelissä pelaajalle näkyviä pistearvoja. Funktio aktivoituu, kun klikataan pelissä näkyviä valintanappeja. Se hakee pelissä tällä hetkellä näkyvään korttiolioon kirjoitetut funktio -ja arvomuuttujat ja hakee taulukosta functionArr muuttujan m mukaisen funktion ja kutsuu sitä muuttujan v arvolla.

```
let pakoon=new Dialogi(14, "Noniin: älä pää
  [{m: 2, v: 2}],[{m:2, v:1}],15,16);
let kiitti=new Dialogi(15, "Ai kiitti! Joo,
  [{m:2, v:2}],[{m: 3, v: 2}],17,18);
let iso=new Dialogi(16, "Ai... joo onhan tä
  [{m: 2, v: 2}], [{m:3, v:2}], 17, 18);
```

Kuva 11: Funktioiden kutsuminen olioon määritellyillä arvoilla.

Kuvassa 11 muuttujalla `m` viitataan metodin järjestysnumeroon taulukossa `functionArr`, muuttujalla `v` viitataan arvoon, jolla kyseistä funktiota kutsutaan. Esimerkiksi `m: 0, v: 25` aiheuttaa pelissä näkyvän pistearvon nousemisen 25:llä.

```
let functionArr = [  
  addPoints, // 0  
  reducePoints, // 1  
  addSocial, // 2  
  reduceSocial, // 3  
  addLife, // 4  
  reduceLife, // 5  
  addMoney, // 6  
  reduceMoney, // 7  
  addAttitude, // 8  
  reduceAttitude, // 9  
  die, //10  
  quitGame, //11  
  changeChapter, //12  
  startEndingChapter, //13  
  setEndingChapterListener, //14  
  setEndingAttitude //15  
]; // array joka pitää sisällään pelin keskeiset funktiot pisteiden muuttamiseksi
```

Kuva 12: Taulukko `functionArr`.

Pelin logiikka ja siten tarinan eteneminen toimii funktioiden avulla. Funktiot on numeroitu kommentteina ohjelmoinnin helpottamiseksi.

Funktion `changeStats` kutsumisen jälkeen piirretään seuraava pelin kuvaruutu funktiolla `draw()`. Tämä funktio pitää sisällään kutsut pelin eri piirrostoimintoihin, jotka ovat tarkoituksella tietyssä järjestyksessä niin, että elementit ilmestyvät ruudulle oikeassa järjestyksessä. Piirtotoiminnot kuten `drawCard`, `drawUI` jne. on jaettu erillisiin osiinsa ohjelmoinnin kokonaisuuden helpottamiseksi. Koodia oli helpompi käsitellä, kun jokainen yksittäinen toiminto oli jaettu omaan funktioonsa. Funktioiden kutsumisen monimutkisuus taas pysyi hallittuna, sillä kaikki funktiokutsut olivat tässä yhdessä funktiossa: `draw()`.

```

881 function draw() {
882     // funktio piirtämiselle, kaikki osat määritellään erikseen omina funktioinaan, alin päällimmäisenä pelinäköymässä
883     ctx.clearRect(0, 0, screenWidth, screenHeight);
884     reduceClickSpeed();
885     if(chapterChange==1){
886         index=0; //huomioi tämä myöhemmin, poista tarvittaessa
887         chapterChange=0; //sama tälle
888     }
889     drawBackground();
890     drawCard();
891     drawUI();
892     drawLife();
893     drawSocial();
894     drawScore();
895     drawMoney();
896     isDrawn=1;
897 }

```

Kuva 13: Kuvankaappaus Draw-funktion sisällöstä.

Pelin piirrosfunktioiden asettaminen Draw-funktion sisälle helpotti koodin monimutkisuuden hallintaa, kun funktioita ei tarvinnut etsiä eri puolilta koodia, vaan ne löytyivät helposti yhdestä paikasta. Tämä helpotti kokonaisuuden hallintaa ja yksittäisiin funktioihin pääsi helposti käsiksi käyttämällä koodieditorin sisäänrakennettua hakutoimitoa.

7.4 Funktion draw toiminta

Funktio draw on keskeinen pelin toiminnalle. Funktio toimii niin, että sen kautta kutsutaan kaikkia grafiikoita piirtäviä toimintoja kerralla joka kerran ruudun näkymää päivitettäessä. Näiden funktioiden varsinainen sisältö on määritelty myöhemmin koodissa.

Funktion alussa käytetään toimintoa `ctx.clearRect`. Tämän avulla pyyhitään HTML-canvas tyhjäksi uuden ruudun piirtämistä varten.

Toiminnolla `reduceClickSpeed` lisätään viivettä valintanappien aktivoitumiseen vahinkotuplaklikkausten välttämiseksi. Kuvassa 13 näkyen rivillä 885 olevassa ehtolauseessa huomiodaan jaksojen eli chapterien vaihtoon liittyvä ongelma, joka korjataan asettamalla tässä kohdassa muuttujan `index` arvoksi 0. Tämä jätettiin koodiin vaikeasti löydettävien bugien korjaamiseksi sekä ehkäisemiseksi. Samassa ehtolauseessa muuttuja `chapterChange:n` arvoksi asetetaan 0, joka estää tämän ehtolauseen aktivoitumisen muutoin kuin kappaleen alussa.

7.5 Funktio drawCard

`drawCard` on funktio, joka piirtää varsinaisen kortin peliruutuun. Funktio alkaa koodissa noin riviltä 2090 ja loppuu noin rivillä 2400. Tämä funktio määrittelee ensin pelissä olevan kortin tyyppin ja toimii sen mukaisesti. Siinä piirretään korttityypin mukaisesti raamit kortille, teksti

niiden sisään ja nappien asettelu. Funktiosta drawCard kutsutaan myös drawInCard-funktiota, joka esitellään seuraavaksi.

```

2216     else if (kortit[index] instanceof Dialogi){ // korttiluokan ollessa Dialogi
2217         if (largeScreenOn==1){ // ehtolause sille, jos pelissä on iso ruutu käytössä
2218             ctx.roundRect(x-320, y - 300, 640, 465, 10, 10, 8); // suorakulmio, itse kortti leveässä ruudussa
2219         }
2220         else if (largeScreenOn==0){ // ehtolause sille, jos pelissä on pieni ruutu käytössä
2221             ctx.roundRect(x-250, y - 260, 500, 420, 10, 10, 8); // suorakulmio, itse kortti kapeassa ruudussa
2222         }
2223         ctx.strokeStyle="white"; // ääriiviivojen värin valinta HTML-canvasille (ctx) piirrettäessä
2224         ctx.fillStyle = "#39a0ca"; // täyttöväri
2225         ctx.fill(); //täytetään värillä, viivan saa käyttämällä ctx.stroke()
2226
2227         ctx.fillStyle = "white"; // tekstin väri
2228         ctx.stroke(); // piirretään HTML-canvasille määritellyt piirrokset
2229     }

```

Kuva 14: Osio drawCard-funktiosta. Tämä osio aktivoituu, kun pelissä olevan kortin luokka on Dialogi.

Kuten kuvan 14 koodissa näkyvissä kommentteissa osoitetaan, funktiossa drawCard on määritetty eri asetukset kuvioiden ruudulle piirtämistä varten. Ruudun kokoa ilmaistaan muuttujan largeScreenOn arvolla, joka voi olla 0 tai 1: 0 pienelle ruudulle, 1 isolle.

7.6 Funktio drawInCard

Tämä funktio on vastuussa pelin kortin sisälle piirrettävistä kuvista, joiksi lukeutuu kortin sisällä oleva maisemakuva ja varsinaisen hahmon kuva. Tämä funktio oli kokonaisuudessaan pitkään drawCard-funktion sisällä, mutta lopulta eriytettiin omaksi funktiokseen koodin suoraviivaistamiseksi ja turhan toiston poistamiseksi koodista. Näin tekemällä saatiin myös korjattua pitkään projektia vaivanneita bugeja, jotka liittyivät siihen, että kortin sisällä näkyvät taustakuvat ja hahmokuvat eivät aina latautuneet.

```

2350
2351 function drawInCard(){// 19.9 yhdistetty eri korttiluokkien kuvien piirtäminen tähän koodin selkeyttämiseksi
2352     function drawActualImage(){ // funktio varsinaisen kuvan piirtämiseksi kortin sisälle
2353         ctx.shadowColor="black";
2354         ctx.shadowBlur=15;
2355         let img=korttiKuvat[index]; // luodaan kuva
2356         if(img.complete){ // jos kuva on latautunut
2357             // console.log("img.complete"); tätä ja monia vastaavia console.log()-toimintoja käytettiin bugien korjaamiseen
2358             ctx.shadowColor="black"; // varjojen väri, aktivoidaan uudestaan tässä
2359             ctx.shadowBlur=15; // varjojen reunojen säätäminen
2360             if(largeScreenOn==0){ // pienen ruudun ollessa aktiivinen
2361                 ctx.drawImage(korttiKuvat[index], x - 125, y - 130, 250, 250); // piirretään varsinainen kuva kortin sisälle
2362                 drawHahmo(); // piirretään hahmo korttikuvan sisälle, tämä funktio on määritelty myöhemmässä kohdassa
2363             }
2364             else if(largeScreenOn==1){ // ison ruudun ollessa aktiivinen
2365                 ctx.drawImage(korttiKuvat[index], x-150, y - 150, 300, 300); // varsinaisen kuvan piirtäminen
2366                 drawHahmo(); // hahmon piirtäminen
2367             }
2368             drawHahmo(); // hahmon piirtäminen varmuudeksi, käytetään korjaamaan vaikeasti havaittava bugi
2369         }
2370         else if(!img.complete){ // jos kuva ei ole latautunut
2371             img.onload=function(){ // kuvan latautuessa tämä blokki aktivoituu
2372                 ctx.shadowBlur=15; //tämä korjasi varjo-ongelman (varjo ei näkynyt ihan ensimmäisessä kuvassa)
2373                 ctx.shadowColor="black";
2374             }
2375             if(largeScreenOn==0){
2376                 ctx.drawImage(korttiKuvat[index], x - 125, y - 130, 250, 250);
2377                 drawHahmo();
2378                 // console.log("hahmoKuvaLadattu");
2379             }
2380             else if(largeScreenOn==1){
2381                 ctx.drawImage(korttiKuvat[index], x-150, y - 175, 300, 300);
2382                 drawHahmo();
2383             }
2384         }
2385     }
2386     drawHahmo();
}

```

Kuva 15: Kuvankaappaus drawInCard-funktiosta

Kuvassa näkyy, mitä funktion drawInCard sisältää. Sisällä on tyyleihin kuten varjoihin vaikuttavia ja suoraan kuvia piirtäviä toimintoja.

7.7 Funktiot callAnimation ja drawAnimations

Funktion callAnimation tarkoituksena on kutsua funktiota drawAnimations. Siihen on myös kirjoitettu sääntöjä sille, miten animaatiot tulisi piirtää, kuten ruudunpäivityksien välisen intervallin pituus ja edellisen animaation pysäyttäminen silloin, kun uuden animaation piirtäminen aloitetaan.

Funktion drawAnimations tarkoituksena on piirtää peliin animaatiot, jotka aktivoituvat valintapainikkeita painettaessa. Tässä funktiossa myös säädetään ajastin, joka säätää animaatioiden näkyvyyttä ja pysäyttää ne niiden poistuttua ruudusta.


```

2900     function drawAnimation() {
2901         ctx2.clearRect(0, 0, canvas2.width, canvas2.height);
2902
2903         if(animationArray[0]==1){
2904             animatePointsGain();
2905             a+=2;
2906             if(a==100){
2907                 animationArray[0]=0;
2908                 a=0;
2909             }
2910         }
2911         if(animationArray[1]==1){
2912             animatePointsLoss();
2913             b+=2;
2914             if(b==100){
2915                 animationArray[1]=0;
2916                 b=0;
2917             }
2918         }
2919         if(animationArray[2]==1){
2920             animateSocialGain();
2921             c+=2;
2922             if(c==100){
2923                 animationArray[2]=0;
2924                 c=0;
2925             }
2926         }

```

Kuva 16: Kuvankaappaus funktiosta drawAnimation. Funktio jatkuu hyvin samannäköisenä pitkään. Jokainen pistekategoria on animoitu omalla funktiollaan pisteiden saannin ja menettämisen osalta.

Funktiossa drawAnimation piirretään varsinaiset animaatiot. Pelin animaatiot ovat yksinkertaisia, väritettyjä plus- ja miinusmerkkejä, jotka leijuvat pelissä näkyvistä pistekategorioista ylöspäin osoittaakseen pisteiden saannit ja menetykset kategorioittain. Näiden toteuttaminen Javascriptilla HTML-canvasille onnistuneesti ei kuitenkaan ollut yksinkertaista. Animaatiot saatiin kuitenkin lopulta toimimaan hyvin. Jokaiselle pelin pistekategorian positiiviselle tai negatiiviselle muutokselle on oma animaationsa, jota kutsutaan erikseen. Tämä vie koodissa paljon tilaa, mutta on helppo tapa hahmottaa koodia kirjoittaessa jokaisen toiminnon ollessa omassa funktiossaan, joista jokainen on asetettu taulukkoon. Näitä funktioita kutsutaan suoraan taulukosta silloin, kun pisteisiin vaikuttavat funktiot kuten addPoints tai addLife aktivoidaan.

7.8 Funktiot changeChapter ja changeKuvat

Peli koostuu viidestä jaksosta, koodissa näitä ilmaistaan muuttujalla chapter. Nykyinen jakso valitaan pelin koodissa muuttujan chapter avulla, jolle annetaan numeroarvo väliltä 0-5. Chapter 0 tarkoittaa introjaksoa, 1 Illan jaksoa, 2 Roihun jaksoa, 3 Laineen jaksoa, 4 Mantanjaksoa ja 5 loppujaksoa. Jakson vaihtoon liittyy kuvien vaihto sekä korttien vaihto. Kuvien vaihto tapahtuu funktiolla changeKuvat, jolle annetaan argumenttina haluttu jakso muuttujan chapter lukuarvona.

```
272 function changeKuvat(chapter){
273     if (chapter==0){
274         // console.log("chapter 0 kuvat")
275         kuvat[0] = { kuva: "images/alkukuva2.png", tkuva: "images/alkukuva.png", hkuva:"" };
276         kuvat[1] = { kuva: "images/alkukuva2.png", tkuva: "images/alkukuva.png", hkuva:"images/hahmot/roihi_pieni_iloinen.png" };
277         kuvat[2] = { kuva: "images/alkukuva2.png", tkuva: "images/alkukuva.png", hkuva:"images/hahmot/roihi_pieni_iloinen.png" };
278         kuvat[3] = { kuva: "images/alkukuva2.png", tkuva: "images/alkukuva.png", hkuva:"images/hahmot/roihi_pieni_iloinen.png" };
279         kuvat[4] = { kuva: "images/alkukuva2.png", tkuva: "images/alkukuva.png", hkuva:"images/hahmot/roihi_pieni_iloinen.png" };
280         kuvat[5] = { kuva: "images/alkukuva2.png", tkuva: "images/alkukuva.png", hkuva:"images/hahmot/roihi_pieni_iloinen.png" };
281     }
282
283     for (let i = 0; i < kuvat.length; i++) {
284         korttiKuvat[i] = new Image();
285         taustaKuvat[i] = new Image();
286         hahmoKuvat[i] = new Image();
287
288         korttiKuvat[i].src = kuvat[i].kuva;
289         taustaKuvat[i].src = kuvat[i].tkuva;
290         hahmoKuvat[i].src = kuvat[i].hkuva;
291     }
}
```

Kuva 17: Kuvankaappaus funktiosta changeKuvat.

Funktio changeKuvat vaihtaa kuvat-taulukon sisältöä niin, että sen sisältö vastaa pelin nykyisen jakson sisältöä. Näin toimimalla saatiin indeksoitua kuvat niin, että niiden järjestysluvut vastasivat korttiolioiden järjestyslukuja. Tämä mahdollisti kuvien käytännöllisen käsittelyn.

```

1006 function changeChapter(jakso) { //muuttaa chapteria
1007   chapter=jakso
1008   index=0; //-----tätä muuttamalla pääsee testatessa suoraan tiettyyn indxiin
1009   changeKorttiStack(jakso);
1010   changeKuvat(jakso);
1011   musiikinValinta(chapter);
1012   changeAvustajat(chapter);
1013   chapterChange=1 //chapterchange 1
1014 }
1015
1016 function startEndingChapter(luku){
1017   endingChapterOn=luku; //arvolla 0 ei olla loppuchapterissa, arvolla 1 ollaan
1018   changeKuvat(5);
1019 }
1020
1021 function setEndingChapterListener(luku){
1022   //defaulttina käytössä, valitun chapterin mukaan kyseinen loppu
1023   if(luku==1){
1024     iltalListener=1;
1025   }
1026   else if(luku==2){
1027     roihulListener=1;
1028   }
1029   else if(luku==3){
1030     lainelListener=1;
1031   }
1032   else if(luku==4){
1033     mantalListener=1;
1034   }
1035 }

```

Kuva 18: Kuvankaappaus funktion `changeChapter` alusta. Tämä on rivimäärältään pisin funktio koodissa.

Pelin varsinainen sisältö on pääosin kirjoitettu funktion `changeChapter` sisälle. Se sisältää varsinaiset korttoliot, joista valitaan jakson mukaisesti oikea lajitelma kortteja. Kortit asetetaan listaan, josta niitä sitten luetaan muuttujan `index` arvon mukaisesti ja tämän jälkeen piirretään varsinainen sisältö. Korttikuvat tulevat omasta kuvalistastaan, joka on valittu funktion `changeKuvat` avulla. Sisältö teksteineen ja kuvineen piirretään funktion `draw` sisällä olevien piirtofunktioiden avulla.

```

1366 else if(chapter==2){
1367   kortit=[];
1368   //ROIHU
1369   let hurisee=new Dialogi(0, "Kuinkas hurisee?", ["Mulla alkoi just kesäloma",""], [(m:2,v:2),(m:0,v:50),(m:14,v:2)],[(m:4,v:2)],1);
1370   let suunnitelma=new Dialogi(1, "loistavaal.\nOnko sulla jotain suunnitelmiä?", ["Itseessäs ei ole", "jotain lomailua meinasin",""],[],2,4);
1371
1372   let loysit=new Dialogi(2, "Mitenkäs löysit tänne?", ["kunhan harhailin",""],[(m:0,v:50)],[],3);
1373   let askelia=new Dialogi(3, "No hyppää kyytiin!", ["Mä en tykkää et mulle kerrotaan mitä mun pitäisi tehdä","näytä tietä sitten vain",""],[(m:2,v:2)],[(m:2,v:2),(m:0,v:50)],6,8);
1374
1375   let kouluvuosi=new Dialogi(4, "Oliko kouluvuosi rankin ikinä?", ["Ei oikeastaan","Aika lailla",""],[(m:0,v:50)],[],5, 52);
1376   let ajateltavaa=new Dialogi(5, "kuulostaa siltä että tykkäät\ropiskella", ["Jos on jotain tarpeellista tietoa niin kuuntelem mielelläni",""],[(m:0,v:50)],[],7);
1377
1378   let asenne=new Dialogi(6, "Mä tykkään sun asenteesta,\nmut halusitko oppia jotain?", ["Mielelläni", "En oikeastaan", ""], [(m:2,v:2),(m:0,v:25)],[],7,26);
1379   ///
1380   let kuuntele=new Dialogi(7, "Hienoa, mulla on paljon näytettävää sulle.", ["Seuraava",""],[(m:0,v:50)],[],8);
1381   let heratyksellö=new Dialogi(8, "Voin opettaa aarekemiassani jotain loistuja.\nmä herään silloin kun muut nukkuu.",["Onkohan toi valvomien kovin terveellistä?",""],[(m:0,v:25),(m:6,v:50)],[],9);
1382   let ketakiinnostaa=new Dialogi(9, "keta kiinnostaa?!", ["No mua",""],[(m:4,v:2)],[],10);
1383   let savua=new Dialogi(10, "Ootsä koskaan maistanut\ntai haistanut savua?", ["Tottakai", "", ""],[],[],11);
1384   let milloinlähtee=new Dialogi(11, "Pidatko siitä?","Riippuu tilanteesta",""],[],[],12);
1385   let eläväsilmälitka=new Dialogi(12, "Ei savua ilman tulta, sanotaan.\nmä osaan syyttää sen.",["No näytspä",""],[],[],13);
1386   let paljonmaksat=new Dialogi(13, "Paijonko maksat?\nEi tää show ilmaista ole",["Annan vitosen", "No en mitään?",""],[(m:7,v:5),(m:0,v:50)],[(m:6,v:5),(m:1,v:50),(m:3,v:2)],14,22);
1387

```

Kuva 19: Kuvankaappaus Roihun jakson sisällön alkuosasta funktion `changeChapter` sisällä.

Kaikki pelin pelaajalle näkyvä tekstisisältö on kirjoitettu korttoliioihin.

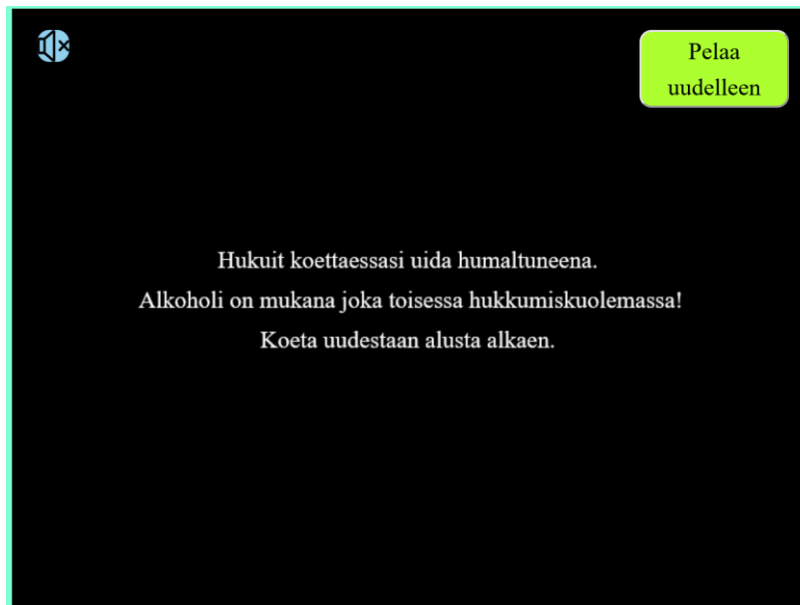
Muuttuja toimii järjestyslukuna, jonka avulla valitaan oikea kortti koko pelin ajan. Korttoliot ovat listassa tismalleen oikeassa järjestyksessä numeroituina, näihin kytketyillä kuvolioilla on kullakin korttoliota vastaava järjestysluku, jolloin erilliset kuva- ja korttoliot kytkyvät osaksi yhteistä kokonaisuutta, joka kokonaisuudessaan näkyy pelaajalle toimivana pelinä grafiikkoineen.

7.9 Funktio die

```
997 function die(){  
998     showDeathScreen();  
999 }
```

Kuva 20: Kuvankaappaus funktiosta die.

Funktion die tarkoituksena on päättää peli ja asettaa kuolemaruutu näkyville. Tämä on tehty omaksi funktiokseen selkeyden takia. Alun perin tähän funktioon oli tarkoitus rakentaa lisää toimintoja, mutta se jäi peliin tällaisenaan.



Kuva 21: Kuolemaruutu.

Kuolemaruutu koostuu tekstielementistä, pelaa uudelleen -napista ja äänien hiljennysnapista. Nämä ovat itsessään yksinkertaisia elementtejä, mutta niiden asettelu sekä pienessä että isossa ruudussa oikein näkyviksi oli työlästä.

```

2664     function showDeathScreen(){
2665         muteDiv.style.left="-40px";
2666         muteDiv.style.top="-30px";
2667         deathScreenOn=true;
2668         stopAnimation();
2669         ctx2.clearRect(0, 0, canvas2.width, canvas2.height);
2670         endingScreenOn=true;
2671
2672         // muteNappi.style.display="none";
2673         if (aanet=="true"){
2674             taustaMusiikki.src("./sounds/sadCaves.mp3");
2675             taustaMusiikki.stop();
2676             let loppuMusiikki=new Sound("./sounds/sadCaves.mp3");
2677             loppuMusiikki.sound.volume=0.3;
2678             loppuMusiikki.play();
2679             console.log("loppumusiikkitässä")
2680         }
2681
2682         // buttons.style.visibility="none";
2683         kylla.style.visibility="none";
2684         ei.style.visibility="none";
2685         valinta.style.visibility="none";
2686
2687         pelaaUudelleenNappi.style.display="block";
2688         console.log("gqegqeegqgeqqg2");
2689         if(largeScreenOn==1){
2690             pelaaUudelleenNappi.style.top="-0px";
2691             muteDiv.style.top="-75px";
2692         }
2693         else if(largeScreenOn==0){
2694             pelaaUudelleenNappi.style.top="-240px";
2695         }
2696
2697         ctx2.textAlign="center";
2698         ctx2.fillStyle="black";
2699         ctx2.fillRect(0,0,1280,720);
2700         ctx2.fillStyle="white";
2701         ctx2.font="24px century-gothic"
2702
2703         ctx2.fillText("Hukuit koettaessasi uida humaltuneena. ", x, y-40);
2704         ctx2.fillText("Alkoholi on mukana joka toisessa hukumiskuolemassa!", x, y);
2705         ctx2.fillText("Koeta uudestaan alusta alkaen. ", x, y+40);
2706     }
2707

```

Kuva 22: Kuvankaappaus funktiosta showDeathScreen, joka piirtää kuolemaruudun.

Kuolemaruutua peliin rakennettaessa ilmeni haasteita sen kanssa, että kuolemaruudun sijaitessa pitkällä pelin sisällössä, sen testaaminen oli vaikeaa. Kuolemaruudun tehokkaammaksi testaamiseksi tehtiin niin, että asetettiin kuolemaruutua kutsuva funktio die pelin alkuun, jotta kuolemaruutua pääsi testaamaan suoraan. Tämän menetelmän avulla saatiin kuolemaruudun elementtien asettelu kuntoon.

7.10 Funktio drawResultScreen

Tämä funktio aktivoituu pelin päättyessä, mikäli pelaaja ei kuole pelissä ennen loppua. Tämä funktio pyyhkii koko ruudun sisällön, pysäyttää pelin toiminnan ja poistaa valintanapit, piirtää pistekategorioita havainnollistavia kuvia ja näyttää pisteet saateteksteineen. Loppuruutuun ilmestyy myös nappi, jonka avulla voidaan kutsua funktiota restartGame ja aloittaa peli alusta.



Kuva 23: Pelin loppuruutu. Pisteluokkia hahmottelevat kuvat ja saatetekstit pelaajan suorituksen mukaan kategorioittain.

Loppuruutufunktio sisältää erilaisia pelin elementtien tyylejä sääteleviä rivejä. Funktio myös aktivoi loppumusiikin ja piilottaa tarpeettomat napit. Funktioon on kirjoitettu pelaajan saamista pisteistä riippuvien tekstien määrittelystä vastaava ehtolause.

```

2708 function drawResultScreen(){
2709     muteDiv.style.left="-55px";
2710     muteDiv.style.top="-40px";
2711     stopAnimation();
2712     ctx.clearRect(0, 0, ctx.width*2, ctx.height*2);
2713     ctx2.clearRect(0, 0, canvas2.width, canvas2.height);
2714     kylla.style.visibility="hidden";
2715     ei.style.visibility="hidden";
2716     valinta.style.visibility="hidden";
2717     endingScreenOn=true;
2718
2719     if (aanet!="false"){
2720         taustaMusiikki.src("./sounds/sadCaves.mp3");
2721         taustaMusiikki.stop();
2722         loppuMusiikki.sound.volume=0.3;
2723         loppuMusiikki.play();
2724     }
2725     kylla.style.display="none";
2726     ei.style.display="none";
2727     valinta.style.display="none";
2728
2729     ctx2.fillStyle="#51d0de";
2730     ctx2.fillRect(0,0,1280,720);
2731     ctx2.fillStyle="black";
2732     ctx2.font="32px century-gothic"
2733
2734     let pisteTulos="";
2735     let sosiaalisuusTulos="";
2736     let elkkuTulos="";
2737     let rahaTulos="";
2738
2739     if (pisteet>1000){
2740         pisteTulos="Mahtava Tulos!";
2741     }
2742     else if (pisteet>500){
2743         pisteTulos="Hyvä Tulos!";
2744     }
2745     else if (pisteet>250){
2746         pisteTulos="Ok tulos";
2747     }
2748     else if (pisteet<250){
2749         pisteTulos="Ei mainitsemisen arvoinen tulos";
2750     }
2751

```

Kuva 24: Kuvankaappaus funktiosta drawResultScreen, joka piirtää loppuruudun.

Loppuruudun rakentamisessa oli samankaltaisia haasteita kuin kuolemaruudun rakentamisessa johtuen loppuruudun sijoittumisesta aivan pelin loppuun. Tässäkin tapauksessa asetettiin loppuruudun näyttävä funktiokutsu heti pelin alkuun, jotta päästiin testaamaan sitä toistuvasti ja näkemään heti koodiin tehtyjen muutosten vaikutukset peliin.

7.11 Funktio restartGame

Tämä funktio aktivoituu loppuruudun pelaa uudelleen -napista. Funktion tarkoituksena on nolata kaikki mahdolliset pelin aikana muutetut muuttujien arvot, jotta peli voidaan aloittaa kokonaan uudestaan.

```
function restartGame(){
  canvas.style.display="none";
  animationLayer.style.display="none";

  if (aanet=="true"){
    taustaMusiikki.stop();
    loppuMusiikki.sound.volume=0.3;
    loppuMusiikki.stop();
  }

  gameOn=false;
  endingScreenOn=false;
  deathScreenOn=false;

  endingChapterOn=0;
  endingListener=0;
  iltalListener=0;
  roihuListener=0;
  laineListener=0;
  mantaListener=0;
  pisteet=0;
  rahat=20;
  elkut=100;
  sosiaalisuus=50;

  iltaAttitude=0;
  roihuAttitude=0;
  laineAttitude=0;
  mantaAttitude=0;
  ctx.clearRect(0,0, screenWidth, screenHeight);
  ctx2.clearRect(0,0, screenWidth, screenHeight);

  kylla.style.visibility="hidden";
  ei.style.visibility="hidden";
  valinta.style.visibility="hidden";
  valikko.style="";
  valikko2.style="";
  alkukuva.style="";
  index=0;
  chapter=0;
  gameEnder=1;
  pelaaUudelleenNappi.removeEventListener("click", restartGame)//poistetaan kyseinen event listener ongelmien ratk
  pelaaUudelleenNappi.style.display="none";
  kylla.removeEventListener("click", yesButtonClicked); //eventlistener napeille
  ei.removeEventListener("click", noButtonClicked);
  valinta.removeEventListener("click", valintaButtonClicked);

  muteDiv.style.left="";
  muteDiv.style.top="";
  muteText.innerHTML="Äänet pois päältä tästä napista";
  muteText2.innerHTML="Jos peli ei näy kokonaan tai näkyy liian pienenä, paina pohjassa CTRL-näppäintä ja vieritä h
}
}
```

Kuva 25: Kuvankaappaus funktiosta restartGame.

Funktio `restartGame` myös asettaa ruudulla näkyvien elementtien kuten alkuruudun opaste-tekstien sijainnin samanlaiseksi kuin ennen pelin aloittamista. Funktioon ajamisen jälkeen kaikki on valmista pelin uudelleenaloittamisen kannalta. Tämän jälkeen pelin suorituksen loputtua näkyy peliruudussa alkuruutu uudelleen, josta pelaaja pääsee halutessaan pelaamaan peliä uudestaan.

7.12 HTML-canvaselle piirtäminen



Kuva 26: Prototyypin tila 29.8.2023.

HTML-canvas-elementissä on sisäänrakennettuna paljon hyödyllisiä ominaisuuksia asioiden piirtämiseksi canvas-elementille. Canvas-elementille piirtämiseen voi vaikuttaa paljon omia funktioita lisäämällä. Pelin JavaScript-koodiin on lisätty mm. Funktio `roundRect()`, jolla voi piirtää suorakulmioita, joiden kulmat ovat pyöristetyt. Rakennettiin funktioon `drawCard()` avustajien piirtämisestä vastuussa olevaan osioon toiminnallisuus, joka mittaa avustajalle syötettyjen tekstirivien leveyden, laittaa ne taulukkoon, järjestää taulukon suurimmasta pienimpään ja palauttaa sitten leveimmän rivin leveyden pikseliarvona. Tätä pikseliarvoa käytetään sitten syöttämällä se funktiolle `roundRect()` ja lisäämällä lukuun 15 pikseliä. Eli jos tekstirivin pituus on esimerkiksi 130 pikseliä, muuttujan `boxWidth` arvoksi tulee 145. Näin tekemällä laatikon leveys muuttuu dynaamisesti tekstirivien pituuden mukaisesti, eikä laatikko jää ylileveäksi.

```

2257 //avustajien pituus on tassa 0:issa
2258     if(avustajat[index]!=undefined){
2259
2260         ctx.shadowColor="white"; //avustajan ja puhelaatikon hohtoefekti
2261         ctx.shadowBlur=15;
2262         ctx.textAlign="left";
2263         let text3=avustajat[index].teksti;
2264         let lines3 = text3.split("\n");
2265
2266         let lineArray=[];
2267         for(let i=0; i<lines3.length; i++){
2268             let lineWidth=ctx.measureText(lines3[i]);
2269             lineArray.push(lineWidth);
2270         }
2271         lineArray.sort( (a, b) => b.width - a.width);
2272         let boxWidth=lineArray[0].width;
2273         console.log(lineArray);

```

Kuva 27: Lisätty osa koodissa.

Tämä avustajan puhekuplan leveyttä säätävä toiminto ei ollut oleellinen pelin toiminnan kannalta, mutta sillä oli iso vaikutus pelin lopulliseen ulkoasuun. Ylisuuret puhekuplat olisivat näyttäneet kömpelöiltä pelin kokonaisilmettä tarkastellessa.

8 Pelin tuottaminen

Tuota- vaiheessa testataan vaihtoehtoja asiakkailta, henkilökunnalla tai muilla sidosryhmillä (Koivisto ym. 2019, 46). Prototyypistä saadun palautteen perusteella päätettiin jatkaa aiemmissa vaiheissa valitun idean kehittämistä ja toteuttaa valintapeli korttimuotoisena tiimin jatkaessa korttien sisällön tuottamista, tarinan muokkaamista ja hahmojen sekä taustojen kuvittamista ohjelmoinnin asettamien reunaehtojen mukaisesti. Testaus nuorilla tehtiin kesällä 2023, kun peli oli valmiimpi.

8.1 Tarina ja hahmokonsepti

Kuinka tarina voidaan yhdistää peliin, on jatkuva ja melko kiistanalainen keskustelu. Kuinka paljon tarinaa on liikaa? Kuinka vähän on liian vähän? Pitäisikö pelin muuttaa tarinaa?

Pitäisikö tarinan sanella pelin kulku? Ei ole yhtä oikeaa vastausta näihin kysymyksiin, mutta se käy selvästi ilmi sekä pelaajien että suunnittelijoiden mielestä, että tarina voi luoda voimakkaan tunteellisia tuloksia. (Fullerton 2008, 41.)

Näitä kysymyksiä pohdittiin tässäkin projektissa ja nuorilta saatu testauspalautte osoitti, että pelin tarina oli sanellut liikaa pelin kulkua, eikä pisteytys toiminut halutulla tavalla. Tämän visuaalisen novellin sankari on nuori itse. Tekemällä hyviä ja terveellisiä valintoja kerryttää pistesaaalista, joka tulee pelin loputtua näkyville lyhyen palautteen kanssa. Tarinassa seikkaillevat myös kuvitellut hahmot, jotka ovat pelaajan ystäviä ja koulukavereita.

Hahmokonseptitaiteilijan työ on luoda hahmoja, joiden erityispiirteet kertovat omaa tarinaansa useilla pienillä elementeillä, jotka kertovat itsessään jo paljon. Hahmot ovat ainutlaatuisia, -kokoisia ja -näköisiä. Hahmojen kautta ja avulla kaikki tarinallinen mediaviihde oikeastaan saa muotonsa. (Artlex.com 2022.)

Pelissä on neljä päähahmoa. ”Hahmot ovat satuhahmoja modernilla vaatetuksella ja lähestymistapa hahmoihin on elementaarinen ja mytologinen” (Vaikkola 2022). Käsikirjoitus perustuu piirroshahmojen ja pelaajan väliseen dialogiin, sillä Vaikkolan mukaan sosiaalisuus vaikuttaa teinien tekemiseen ja puhe kuvaa hyvin sitä. Jokaisella piirroshahmolla on pelissä oma jaksosensa, jossa myös käsitellään tiettyjä teemoja. Teemat ovat: alkoholi, muut päihteet, terveelliset elämäntavat (ravitsemus, liikunta, uni), virushepatiitit. Asiakkaan listaus sisällöistä on liitteessä 2 tämän raportin lopussa.

Tarinoita kerrotaan, koska on jotakin sanottavaa. Tässä tapauksessa sanoma on munuaisten ja maksan terveys sekä siihen vaikuttavat elämäntapavalinnat. Tarinaan on yritetty saada mukaan mahdollisimman paljon asiakkaan listaamia sisältöjä, mutta tarinan elävöittämiseksi on käytetty muunlaistakin sisältöä pohjautuen tausta- ja esitutkimukseen, joissa nuoret kertoivat kokemuksistaan ja ajatuksistaan. Nuorten elämä on erilaisissa ympäristöissä kuten vaikkapa kaupungeissa ja maaseudulla. Tavoitteena on saada pelaaja pelaamaan peliä useamman kerran myös siksi, että haarautuvien polkujen varrella on eri sisältöjä. Monessa haarassa tarina sekä dialogi on kuitenkin kirjoitettu niin, että tärkeä terveydellinen sisältö ilmenee joka tapauksessa.

Tarinaa kehitettiin Miro-yhteiskehittämislustalla. Peli on modulaarinen, eli paloista koostuva ja sitä on helppo muuntaa, sekä siihen on helppo lisätä osia. Sitä voisi myös laajentaa käsittelemään vaikkapa moraalisia kysymyksiä ja valintoja liittyen esimerkiksi varastamiseen. Pelin eri tarinavaihtoehdot eli skenaariot vaikuttavat pelin lopputulokseen. Skenaario on alun perin termi, jota käytetään teatterin ja elokuvan tekemisen yhteydessä ja joka tarkoittaa näytelmän toiminnallista käsikirjoitusta (Rubin, 2023). Skenaariot ja skenaariotyöskentely ovat laaja kokonaisuus, jota tässä ei käsitellä tämän enempää muutoin kuin toteamalla niiden olevan

hyvin olennainen osa yhteiskuntaa, liiketoimintaa ja organisaatioita useilla tasoilla myös esimerkiksi asiakaspalvelussa.



Kuvio 11: Ote Roihun kappaleesta Mirossa skenaarioineen.

Fantasiahahmojen kautta sisältö saadaan peliin luontevasti ja jokaisella hahmolla on oma lunksa, jossa käsitellään tiettyjä terveyden ja hyvinvointiin liittyviä teemoja. Pelaaja asetetaan näin arkisiin tilanteisiin ja tapahtumiin, joissa hänen on tehtävä valintoja, kuten oikeassakin elämässä. ”Monestihan on niin, että oikeassakin elämässä tilanteet usein tulevat eteen ja asiat tuntuvat vain tapahtuvan (Vaikkola 2022).” Tarina haarautuu muuttujien mukaisesti.

Graafikon tekemät piirroshahmot on piirretty piirtopöydällä käyttäen maksutonta Autodesk SketchBook -sovellusta. Muitakin vastaavia ohjelmia löytyy omilla erityispiirteillään. Kaikki graafikon luomiin kuviin liittyvät käyttöoikeudet ja muut tekijänoikeuksiin liittyvät asiat on sovittu Munuais- ja maksaliiton palvelupäällikön sekä graafikon itsensä kanssa.

8.2 Tekijänoikeudet

Ilman tekijänoikeutta kulttuurin ja viihteen tekeminen ja tuottaminen ei olisi kannattavaa, sillä kuka tahansa voisi omia kenen tahansa työn itselleen ja hyötyä rahallisesti toisen työn tuloksesta. Tekijänoikeus siis mahdollistaa sen, että meillä on ja meille syntyy vastaisuudessakin uutta kiinnostavaa luettavaa, katsottavaa ja kuunneltavaa. Tekijänoikeus suojaa luovaa työtä ja sen tekijöitä. Se varmistaa luovuuden jatkumisen ja tekijöiden toimeentulon. (Tekijänoikeus.fi 2023.)

Tekijänoikeuslain (446/1995) 1 luvun 1 §:ssä määritellään, että sillä, joka on luonut kirjallisen tai taiteellisen teoksen, on tekijänoikeus teokseen, olkoonpa se kaunokirjallinen tai selittävä kirjallinen tai suullinen esitys, sävellys- tai näyttämöteos, elokuvateos, valokuvateos tai muu kuvataiteen teos, rakennustaiteen, taidekäsityön tai taideteollisuuden tuote taikka ilmetköönpä se muulla tavalla.

Suomen tekijänoikeuslaki (34/1991) säätelee myös, että kirjallisena teoksena pidetään myös karttaa sekä muuta selittävää piirustusta tai graafista taikka plastillisesti muotoiltua teosta sekä tietokoneohjelmaa. Tähän peliin kuvitettujen piirroshahmojen tekijänoikeudet löytyvät liitteestä 3 tämän raportin lopussa.

8.3 Käsikirjoitus ja luvut

Pelin luvut on kirjoitettu dialogin muotoon ja asiasisältö tulee esiin sekä valintojen yhteydessä, että pelin sisäisissä keskusteluissa itsessään. Tarinaan on pyritty sisällyttämään paljon valintoja, jotta nuori kokee klikkailemalla mahdollisimman paljon osallistuvansa tarinaan. Osa valinnoista on pisteytettyjä ja osa ei ole. Luvut haarautuvine skenaarioineen ovat vuokaavioita.

Vuokaavio on graafinen tapa dokumentoida toimintosarja, vuokaaviot toimivat kuvallisena välineenä kommunikoida henkilöltä toiselle tapahtumien tai toimien aikajärjestyksestä (ANSI 2023). Iltan jakso on kuvittajan/dramaturgin kirjoittama malliesimerkki, muut jaksot on kirjoittanut Palveluliiketoiminnan johtamisen opiskelija. Iltan jakso poikkeaa hieman muista rakenteeltaan.



Kuvio 12: Iltan kappaleen skenaarioita Mirossa.

Tarina alkaa Ilta Ilmattaren jaksosta, joka käsittelee sosiaalista mediaa ja alkoholia. Pelaaja saa useita kertoja mahdollisuuden ottaa alkoholia tai kieltäytyä sen nauttimisesta. Seuraukset ovat valintojen mukaisia. Menet bileisiin ja törmäät aurinkoiseen Iltaan, joka on aika humalaisessa tilassa. Ilta innostuu menemään uimaan altaalle takapihalle ja humalassa uiminen on vaarallista. Estätkö häntä menemästä uimaan, vai pysytkö poissa alkoholista ja pysyt uima-vahtina?

Vaihtoehtoisena aloituksena on tavata Lohikäärme Roihu. Ytimen kaupunkialueella pyörii uskaliasta näyttävän näköinen Roihu, joka osaa tuottaa tulta ja savua. Tässä kappaleessa käsitellään pelaamista, unta ja myös sähkötupakointia. Kesäkurssilla Roihu tupakoi saadakseen stressiään ja perheensä aiheuttamia paineita pois. Hän haluaa opastaa myös pelaajaa käsittelemään stressiä esimerkiksi hengittelemällä. Terveydelle pahempi juttu on, jos valitsetkin tupakan polttamisen. Kurssi magiasta onkin luultua haastavampi, mutta selvität kuitenkin tiesi ulos Ytimestä ja siirryt seuraavaan lukuun.

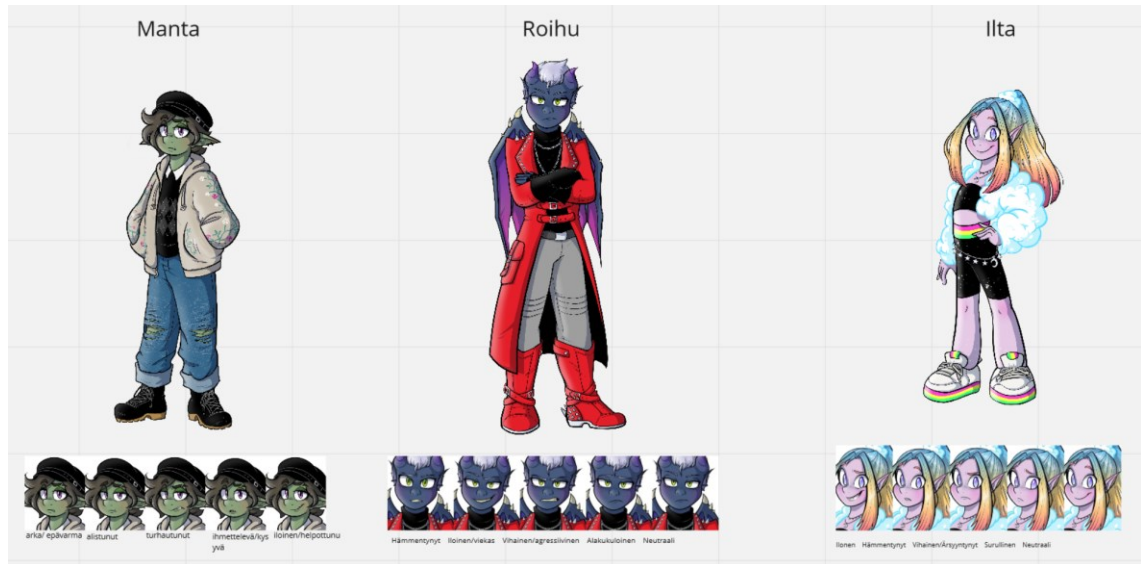
Seikkailu etenee Hyvinvointialueelle. Laine Vetehinen on nuori ja pohdiskeleva tyyppi. Hänen kauttaan käsitellään nuoren omaa sairautta, liikuntaa, ravitsemusta ja seksiä sekä siihen liittyviä terveystarpeita. Hyvinvointiyksikkö on luonnon suuressa, rauhallisessa ja kauniissa kosteikossa etenevä monihaarainen polku, jonka varrella kohdataan Laineen lisäksi Vetehisten perhettä ja ystäviä. Rannalla evästellään ja kalastellaan Laineen kanssa, rauhallinen sekä mukava Laine on urheilullinen. Hän pelkää paljon epäterveellistä ruokaa ja näännyttää itseään, jotta pysyy uimakunnossa, vaikka asia ei näin ole, koska Laine kyllä tietää osaavansa uida ja vie pelaajankin mielellään mukanaan kivoihin vedenalaisiin paikkoihin.

Kesä on edennyt hyvin, vietetään juhannusta ja metsässä töissä oleva Manta Menninkäinen on ravintolatyöntekijä ja tiskijukka. Hänen kauttaan läpikäydään mm. muihin pähteisiin, lääkkeisiin ja luontaistuotteisiin liittyviä vaaroja. Manta johdattelee pelaajan tekemään valintoja liittyen huumausaineisiin. Kappaleen pääsisältöinä ovat myös hepatiitit A, B ja C. Manta on huomannut kesätöissä, että työtoveri on ehkä käynyt roskakatoksessa piikittämässä, kerrotko asiasta esimiehelle vai käytkö sanomassa kaverille varoituksen sanan?

Fantasiapelissä on mahdollisuus käsitellä asioita eri tavoin, kuin jos ne esitettäisiin tosiasioina. Pelissä kristallipallo kuvaa älypuhelinta, magian kurssi tarkoittaa koodauskurssia sekä maksa- ja munuaissairauksista puhutaan yleisemmin tauteina, jolloin ne sopivat paremmin fantasiamaailmaan, mutta ovat kuitenkin ymmärrettävissä avustajankin puheista todellisiksi sairauksiksi. Peli on kehitetty nuorille, joten siitä haluttiin tehdä seikkailullinen, hauska ja jännittävä. Suomen kielellä leikitellään pelissä jonkin verran ja siitä on tehty jopa runollista. Aluksi hyvinkin puhekielisenä tehty dialogi muuttui enemmän kirjakielisemmäksi työstämisen myötä, koska kyseessä on kuitenkin opetuspelejä, jota voidaan käyttää myös kouluissa.

Suurin osa oppaan sanoista on faktapohjaisia munuais- ja maksaliiton sekä sidosryhmien ja järjestöjen verkkosivuilta otettuja suoria lainauksia, jolloin lähde on mainittu. Todellisuus ja fantasiamaailma limittyvät pelissä toisiinsa. Kuvakäsikirjoitus tehtiin Stable Diffusion -tekoälyn luomia taustakuvia käyttämällä Miroon niin, että niitä mm. laitettiin skenaarioissa sopiviin kohtiin. Tarinan muokausvaiheissa kuvat kommunikoivat joissain tilanteissa tarinan tekstin ja dialogin kanssa sekä myös päinvastoin toisiaan kehittäen. Käsikirjoituksessa käytettiin

myös hahmojen eri ilmeitä kuvina apuna ja tukena dialogia keksittäessä, sillä ilmeillä on kommunikoinnissa valtavan suuri merkitys.



Kuva 28: Manta, Roihu- ja Ilta -hahmot ja niiden ilmeet, joita käytettiin tunnetilojen ilmaisuun pelissä.

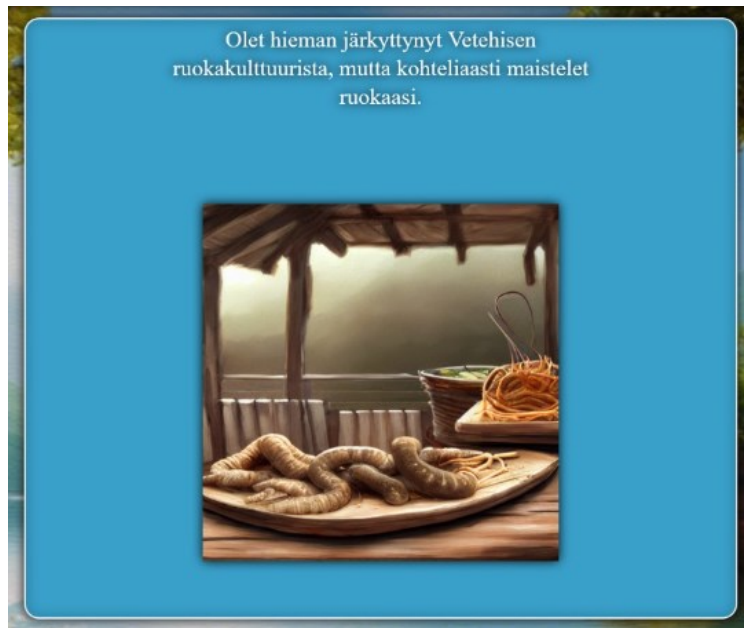
Hahmojen ilmeet vaihtuvat koko pelin ajan dialogin mukaisesti. Näin saatiin elävöitettyä pelin tarinankerrontaa merkittävästi, yksinkertaisella ja toistettavalla tavalla.



Kuva 29: Laine-hahmo ensimmäisessä konseptissa.

Stable Diffusion -tekoälykuvageneraattorin avulla on luotu pelin korttien sisällä olevat taustakuvat sekä korttien takana olevat taustakuvat. Ohjelma toimii Windowsissa Conda-komentoriviä käyttämällä. Conda on erillinen komentorivi Pythonin käyttämiseksi, jolla tämä tekoälykuvageneraattori on koodattu. Stable Diffusion toimii käyttäjän näkökulmasta niin, että ohjelmalle syötetään kehote, niin sanottu prompti, johon kuvaillaan englanniksi vapaasti valittavia avainsanoja käyttämällä mitä kuvaan halutaan, tekoäly sitten luo käsityksensä tästä kuvauksesta.

Kehotteeseen kuuluu sen tekniset osat, avainsanat, toivotut tyylit, parametrit ja kohdekansio, johon kuvien halutaan tallentuvan. Stable Diffusion on raportin kirjoittamisen aikaan maksuton, sen käyttö on luvallista monipuolisesti ei-haitallisiin tarkoituksiin, joihin tämä projekti lukeutuu (Stability AI 2023).



Kuva 30: Kuva madoista Stable Diffusionilla tehtynä.

Ylläolevassa kuvassa 30 näkyy 26.9.2023 Stable Diffusion -tekoälyllä luotu kuva madoista pelissä, kehote kokonaisuudessaan: `python scripts/txt2img.py --prompt "Tasty worms on a wooden table inside a fantasy fishing cottage home, summer, realistic, elegant, highly detailed, digital painting, artstation, concept art, matte, sharp focus, illustration" --plms --ddim_steps 50 --n_iter 5 --n_samples 1 --outdir D:\AI3`

8.4 Pelin markkinointi ja testaus

Vaikka pelin tekijät testasivatkin peliä kymmeniä kertoja sitä tehdessään, on kuitenkin niin, että varsinkin projektin loppuvaiheessa pelin tekemisessä korostuu se, että pelin tekeminen on paljon muuta kuin vain hauskaa pelaamista. Se on edelleen työtä ja vaatii aktiivista tekemistä kuten viestintää, ohjelmointia ja lakkaamatonta innostuksen ylläpitämistä tiimissä. On myös todella tärkeää opetella muitakin taitoja sen lisäksi, että vain tykkää peleistä (Fullerton 2008, 5). Pelin tekijöiden oman jatkuvan testaamisen lisäksi demoa testasivat lääketieteen ammattilaiset, kokemusasiantuntijat, asiakkaat ja sidosryhmät sekä kohderyhmän edustajat.

Pelin demo esiteltiin maksatyöryhmän kokouksessa lääketieteen ammattilaisille sekä kokemusasiantuntijoille 13.6.2023 klo 15.15. Testauskutsut lähetettiin 20.6.2023 kohderyhmän edustajille sekä asiakkaalle ja sidosryhmille. Pelin testaukseen oli alun perin aikaa 13.8.2023 asti, jonka jälkeen palaute purettiin ja sitä hyödynnettiin vielä pelin loppukehityksessä. Pelin testausaikaa pidennettiin hieman syyskuulle. Palautteen käyttö oli tärkeässä osassa pelin sisällön viimeistelyssä.

Taulukko 3: Toteutunut aikataulu

syksy 2022	talvi 22-23	kevät 2023	kesä 2023	syksy 2023
aloitus 24.8	tarina/ sisältö	demo valmis	testaus asiakkaalla, kohderyhmällä sekä ammattilaisilla ja asiantuntijoilla	palautteen perusteella ohjelmoidut muutokset
ideointi ja ohjelmointi	hahmot valmiit	musiikki valmis		Opponointi
prototyyppi valmis	chapterit, skenaariot	Nimi: Fantasia-kesäloma!		Raportin palautus
Hahmoluonnoksia/ hahmokonsepti	ohjelmointia	ohjelmointia		27.9 seminaari/ pelin julkistaminen

”Parhaat oivallukset ja konseptit vetoavat yli kohderyhmärajojen - niiden löytäminen ja luominen vaatii syvällistä ihmisyyden ymmärtämistä. Paras ja vaikuttavin markkinointi perustuu syvälliseen suunnittelutyöhön.” (Markkinointiliitto 2023.)

Testaus tehtiin yli kohderyhmärajojen ja palautteeseen perustuvat muutokset toteutettiin. Saimme mm. 84 sivua muutosehdotuksia asiakkaalta. Tämä koostui kuvankaappauksista ja tekstistä. Lisäksi asiakas oli tehnyt laatimiemme kysymystemme perusteella Webropol-kyselyn, joihin kohderyhmän nuoret vastasivat. Kysely lähetettiin asiakkaan toimesta sekä sidoryhmien välityksellä usealle yläkouluikäisille nuorille, joista kolme vastasi kyselyyn. Kysymykset löytyvät liitteenä.

Testaukseen annettiin aikaa kesän 2023 yli 01.06.2023 - 01.08.2023. Kaikille workshoppiin osallistuneille nuorille lähetettiin linkki peliin ja kyselyyn. Vastauksista saadulla tiedolla muokattiin peliä ehdotusten mukaisesti. Testaukset ja palautteet on tehty nimettömästi ja käyttäen Webropolin kyselylomakepohjaa, Webropol on kyselytutkimustyökalu. Se on työkalu tilanteisiin, joissa tarvitaan vastauksia ja tietoa kohderyhmästä, tässä tapauksessa yläkoululaisista ja nuorista.

Mitä ajattelet pelistä yleensä?

Vastaajien määrä: 3

Vastaukset
Peli on ihan okei. Sen musiikkimaailma ja ääniefektit ovat tosi kauniita ja hienoja. Myös pelissä olevat hahmot ovat hienosti piirrettyjä. Pelistä kuitenkin ei saa paljoo irti opetuksen suhteen eikä tule sellaista tunnetta että pisteet ollenkaan vaikuttaisivat peliin, joten on ihan sama jos saa huonot tai hyvät pisteet.
Tiedoksi testaajat olivat 13 ja 14 vuotta täyttäviä tyttöjä. Ihan mielenkiintoinen ja erilainen.
Vaikutti mielenkiintoiselle va idea näkyy hyvin

Kuva 31: Vastaajien avoin palaute pelistä. Tällä tavoin saatiin osviittaa suoraan nuorilta.

Avoimen palautteen avulla saatiin tieto siitä, millaisena pelaajat kokevat pelin. Palautteen pohjalta kehitettiin pelin sisältöä suoraan. Vaikka palautetta ei saatu suurelta määrältä testaa- jia, jo pienikin testaajien määrä auttoi kehitysprosessissa merkittävästi.

Mitä lisäisit tai muuttaisit?

Vastaajien määrä: 2

Vastaukset
Lisäisin lisää terveyteen liittyvää sisältöä ja muuttaisin hieman sitä, miltä peli näyttää. Peli on hieman liian suuri joten monivalintoihin pitää joka kerta scrollata alas ja se voi muutamana kerran jälkeen alkaa vaikuttamaan ihan pelin pelaamiseen ja kokemukseen. Värimaailmaakin voisi hieman muokata siten, että sävyt eivät olisi niin ristiriidossa toistensa kanssa ja peliä olisi ylipäättänsä mukavempi katsoa ja pelata. Pelin teksti ja fontti voisi myös olla hieman selkeämpää.
A-, B-, C-hepatiitti oli vaikea kohta. Haluatko oireet oli hämmentävä kysymys. Sitä ei ymmärretty ilman, että aikuinen selitti, mihin ne vaihtoehdot A, B ja C viittaa.

Kuva 32: Kehitys- ja muokkausideoita testaa- jilta.

Kyselyssä käytettiin suurimmilta osin vapaa sana -tyyppisiä avoimia kysymyksiä, näin testaa- jat avasivat mielipiteitään ja saatiin laajempia vastauksia kuin mitä suljetuista kysymyksistä ol- taitiin saatu.

Kuivale lyhyesti pelikokemustasi

Vastaajien määrä: 2

Vastaukset
Pelikokemukseni oli ihan mukava, mutta jätti muutaman kerran hämmentyneeksi ja pisteet eivät oikein lisänneet mitään peliin. Pelin hahmot olivat esteettisesti kivoja.
Mantan kanssa muuttaminen oli hämmentävä loppuratkaisu.

Vapaa sana

Vastaajien määrä: 2

Vastaukset
ihan kiva
Ihan kiva. Saattaisin pelata peliä ilman koulutuntia ja sitä, että opettaja tunnilla tai joku muu aikuinen sanoo, että pelaapas tätä. Mutta jotta pelaisin, pelin pitäisi tulla itsestään vastaan, siis olla helposti löydettävissä ilman hakemista.

Kuva 33: Lisää pelitestauksen kommentteja

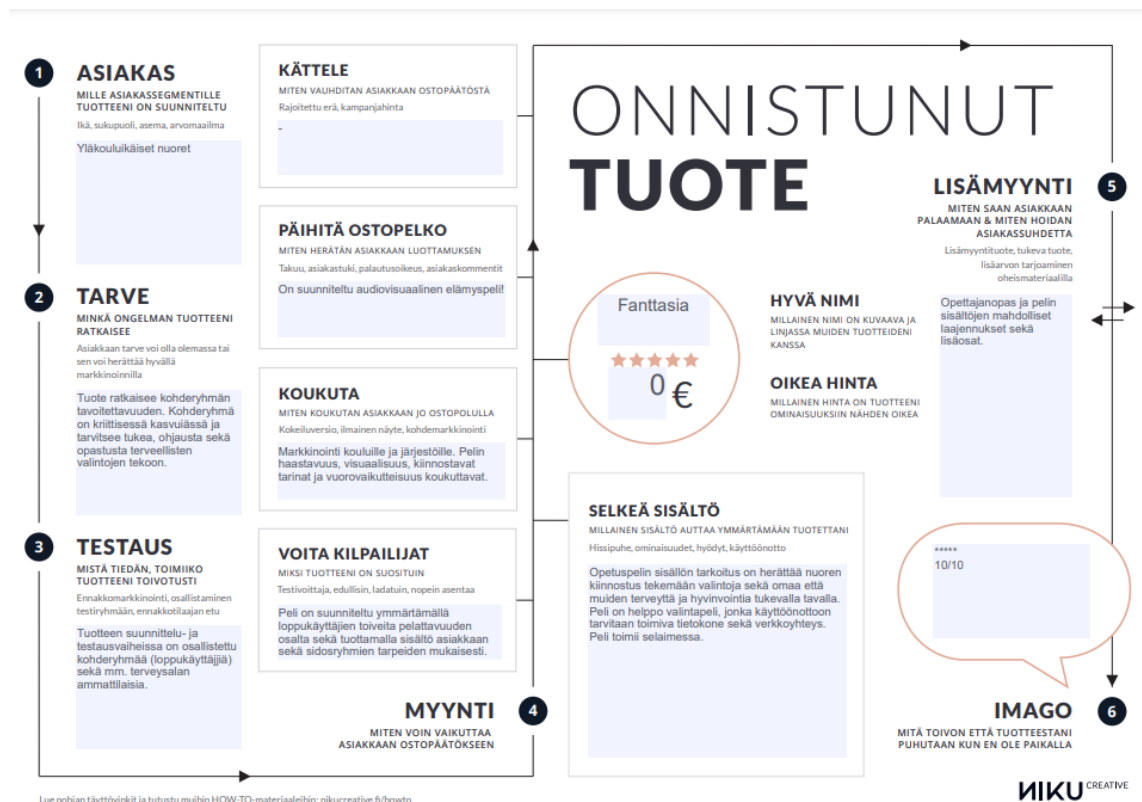
Kuvassa 33 näkyvä kommentti hepattiesta selkeytti kyseisen kohdan suunnittelua pelissä merkittävästi. Sisältö välittyi vaihtelevasti läpi pelistä saamamme palautteen mukaan. Informaation perusteella muokattiin peliä, jotta opettava materiaali olisi havainnollisempaa.

Asiakkaalta ja sidosryhmiltä palautetta tuli lähinnä raha-pistekategorian tarpeenmukaisuudesta, mikä on ymmärrettävää terveyspelin kehittämisen näkökulmasta. Lisäksi testaajat olivat huomanneet paljon bugeja. Palautetta tuli myös tarinankulun lisäksi dialogista liittyen pelaajan valintavaihtoehtojen puuttumiseen tietyissä kohdissa tai tekstin puuttumisesta kokonaan, jolloin pelaajan sanojen tilalla lukee: seuraava, tai näppäimessä on kolme pistettä, mitä painamalla kortti vaihtuu seuraavaan. Myös tietoisuutta kaivattiin enemmän ja niistä pääsyä suoraan tietolähteeseen. Klikattavien linkkien suora koodaaminen peliin olisi kuitenkin voinut rikkoa pelin rakennetta koodissa, joten sitä ei tehty. Lisäksi linkkien osalta on mahdollista, että linkit rikkoutuvat ajan myötä sisällön poistuessa, joten tästäkin syystä klikattavien linkkien poisjättämistä pidettiin loogisena ratkaisuna. Pelissä näkyvät avustajat ja niiden puhekuplat tehtiin testausvaiheen jälkeen näkyvämmäksi lisäämällä niihin hohdetta. Pelin kuva-ruudun skaalautuvuutta ja kokoa säädettiin palautteen perusteella.

Pelin tarinan muuttaminen ei tässäkään projektin vaiheessa ollut enää helppoa, koska korttien numeroituja paikkoja ei sopinut enää vaihdella, muuten pelin rakenne olisi hajonnut ja tämän korjaaminen olisi vaatinut merkittävää lisätyötä. Tämän takia muutokset tarinaan ja pelilogiikkaan oli kirjoitettava niin, että korttien numerjärjestys pysyisi samana. Tämä oli haastavaa, vaikka ideoita tarinan muokkaamiseen esimerkiksi rahaan liittyen tuli kyllä vielä projektin loppuvaiheessakin. Lohikäärmeellä voisi esimerkiksi olla luolassaan iso kasa aarteita, kuten lohikäärmeillä usein on. Rahakategoriaa voitaisiin kerryttää ja siihen liittyvää

kerrontaa lisätä, kunhan olisi selvää, mitä sillä tekee pelissä, kuten ostaa vaikkapa vaatteita, purjevene tai lähteä matkalle. Myös ulkomaanmatkalla piilevät hepatiittiriskit olisi ollut melko helppo kirjoittaa tähän liittyen, kuten joissain aiemmissa tarinaluonnoksissa olikin. Ensimmäinen prototyyppi, joka esiteltiin syksyllä 2022, oli hyvin yksinkertainen verrattuna toiseen versioon pelistä, joka sisälsi kaikki jaksot hahmoineen ja musiikkeineen.

Palautteen pohjalta tehtiin myös 0.3 sekunnin ajastin, joka estää vahingossa tuplaklikkaamisen kortin yli ja täten vahingossa tehdyt väärät valinnat pelissä. Lisäksi tekstien ja pelin käyttöliittymän grafiikoita tarkennettiin, koska saimme palautetta, että teksti voisi olla selkeämpää. Pelin sisäinen renderöintiresoluutio tuplattiin, joka auttoi tähän asiaan ja paransi kuvanlaatua ja tekstin laatua merkittävästi. Valintanäppäimet taustoitettiin valoisan vihreällä, jotta valintapelissä olennainen eli valinnat korostuisivat. Renderöinnin avulla voidaan tehostaa myös myyntiä esittelemällä tuotteita ja palveluita, virtuaalitodellisuuden mahdollisuudet ovat laajat palvelualoilla (Northman VR 2023). Munuais- ja maksaliitto huolehti pelin markkinoinnista sidosryhmiensä kanssa. Peli löytyy liiton nettisivuilta.



Kuva 34: Tuotekortti.

Tuotekorttia voi helposti ja selvästi käyttää tuotteen myynnissä sekä organisaation sisällä palvelun suunnittelussa ja toteutuksessa. Siinä kuvataan mm. palvelun sisältö ja palvelulupaus, eli mitä asiakas tuotteesta saa. (Palma 2023.)

8.5 Tiedonhaku ohjelmointia varten

Peliä ohjelmoitaessa tietoa haettiin runsaasti Googlen avulla. Lähteinä toimi mm. W3Schools (W3Schools 2023), GeeksForGeeks (GeeksForGeeks 2023), Stack Overflow (Stack Overflow 2023), BobbyHadz (Borislav Hadzhiev 2023) ja muita teknisiä ohjelmointiopastus sivustoja. Ongelmanratkaisuun sekä tiedonhakuun käytettiin lisäksi GPT-3.5 ja GPT-4-tekoälyjä, joskin näiden käyttö aloitettiin keväällä 2023 ohjelman perusrungon ja varsinaisen logiikan jo muodostuttua ja ison osan sisällöstä ollessa jo valmiina. ChatGPT:tä ei ole käytetty koodin suoraan kopioimiseen ja liittämiseen suurissa paloissa, mutta sille on syötetty ongelmia ja sen ehdottamia ratkaisuita on useassa kohdassa sovellettu ja jätetty lopullisiksi ratkaisuiksi. Ohjelmointiprosessin edetessä todettiin, että ChatGPT on todella hyvä työkalu yksittäisten ongelmien ratkaisemiseksi koodissa. Pelkällä ChatGPT:n kaltaisella tekoälyllä projektia ei olisi kuitenkaan voinut rakentaa.

Googlea käytettiin tiedonhakuun projektin jokaisessa vaiheessa, ilman Googlea tai vastaavaa hakukonetta ohjelmointi ja ongelmanratkaisu olisi ollut hyvin erilaista ja merkittävästi hitaampaa. Myös Mozillan JavaScript-ohjeita (Mozilla 2023c) käytettiin tiedonhakuun.

8.6 Ohjelmointikielät

Nettisivu, jolla itse peli sijaitsee, on tehty HTML ja CSS-kieliä käyttämällä. Nettisivun elementit on rakennettu HTML:llä ja näiden asettelu ja väriyty on toteutettu CSS-kielillä. HTML-koodissa on asetettu sivulle Canvas-elementtejä päällekkäin kerrostetusti niin, että alemmalle kerrokselle piirretään pelin tekstit ja kuvat, ylemmälle kerrokselle piirretään animaatiot. Canvas-elementeille piirtäminen tapahtuu kokonaan JavaScript-koodista käsin. JavaScript-koodi on liitetty HTML-tiedoston kautta erillisestä tiedostosta osaksi kokonaisuutta. Canvas-elementtien kohdalle on myös asetettu pelin nappulat: aloita-nappi, äänen hiljennysnappi, pelaa uudelleen -nappi ja itse pelin valintanapit: kyllä, valinta, ei.

Pelin logiikka on tehty JavaScriptillä, olio-ohjelmoinnin tekniikoita hyödyntämällä. JavaScriptiin päädyttiin sen helpon jaettavuuden takia: peliä voi pelata suoraan verkkoselaimesta, eikä pelaajan tarvitse asentaa tietokoneelle mitään. JavaScript on myös kielenä hyvä yksinkertaisten pelien kehitykseen, ja sillä koodatut projektit ovat julkaistavissa nettiin helposti. Koodieditorina käytettiin Microsoftin Visual Studio Codea. Projekti on tehty kokonaisuudessaan iteratiivisesti: kaikkea ei tehty kerralla alusta loppuun, vaan pelistä koostettiin toimiva versio, jonka pohjalta lähdettiin kehittämään seuraavaa versiota. Tämä toteutui myös teknisten aspektien osalta.

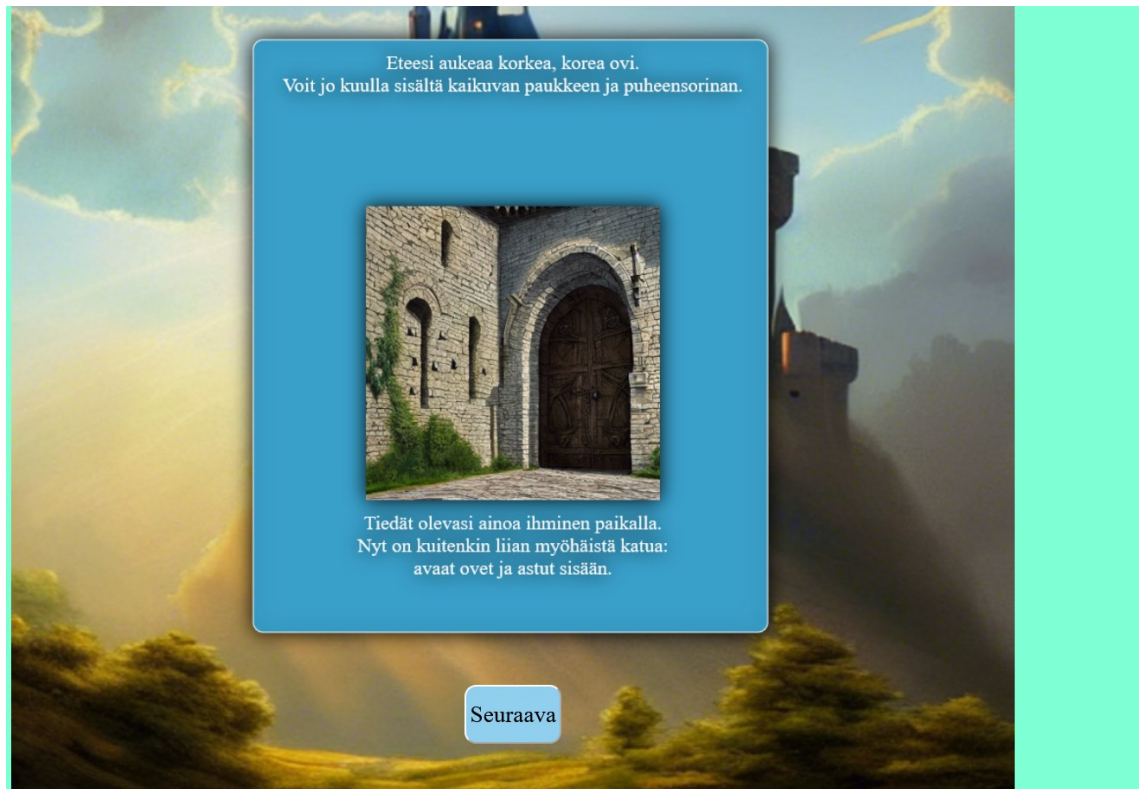
```

index.html # styles.css JS indexjs x
JS indexjs > ...
760   hahmoKuvat[i].src = kuvat[i].hkuva;
761   }
762   }
763
764   function changeKorttiStack(chapter) {
765
766     // tehdään kortteja, olioliat parametreina kohdille kyllä, ei, seuraavaKyllä ja seuraavaEi, (seuraavaValinta) m=metodi, v=vaikutus, saatt
767
768     if (chapter==0){
769       let valikko=new Dialogi(0, "Minkä chapterin valitset?", ["Ilta", "Roihu", "Laine"], [{m:12, v:1}], [{m:12, v:2}], 0, 0, [{m:12, v:3}],0);
770       kortit[0]=valikko
771     }
772
773     if(chapter==1){
774       //ILTA
775       kortit=[];
776       let pilvilinna = new tkortti(0, "Toivottavasti sinulla ei ole korkean paikan kammoa:\nIltan kotiovelle pitää nimittäin ottaa kärry,\njoka nouse
777       [],["Seuraava"], 1 );
778       let ovi = new tkortti(1, "Eteesi aukeaa korkea, korea ovi.\nVoit jo kuulla sisältä kaikuvan paukkeen ja puheensorinan.", "Tiedät olevasi ainoa
779       [],["Seuraava"], 2);
780       let marmori = new tkortti(2, "Astut korkeaan, valkeilla marmorilaatoilla varustettuun eteiseen.\nTalo on täynnä nuoria, musiikki jyskyttää ja j
781       [],["Seuraava"], 3);
782       let ilta=new Kortti(3, "Kato,\nMeiän Nipa Normaali pääsi paikalle!","", ["Seuraava", "none", ""],
783       [],4,4);
784       let ilta2=new Kortti(4, "Se on Ilta, bileiden emäntä!\nHänellä on kädessään kertakäyttömuki\njosta hän huikkaa jotakin", "Alkoholin tuoksu vahv
785       [],5,5);
786
787       let ilta3=new Kortti(5, "No mikäs meisinki?,"Otatko sä tätä\njuotavaa?", ["Otan", "En", ""],
788       [{m: 2, v: 2},{m: 5, v: 2}, {m: 8, v:1}],[{m: 2, v: 2}],8,7);
789       let iltahumalassa=new tkortti(6, "Ilta on hieman humalassa, ja sinua hermostuttaa kieltäytyä.\nMoni haluaisi olla hänen ystävänsä, Iltan vaikut
790       [],["Seuraava"], 6);
791
792       let enjuo=new Kortti(7, "Ai, sä tulit mun bileisiin\netkä ees maista mun ite sekoittamaa juhla juomaa?", "Mä oon saanu tästä paljon kehuja tiätkö
793       [],9,9);
794       let juo=new Kortti(8, "Tää on mun omatekemää\njuomaa jossa on...\nno siis vähän kaikkee.", "Salanen resepti. Maista niin tiiät.", ["Seuraava",
795       [],10,10);
796

```

Kuva 35: Jakson aktiivisten korttiolioiden määrittely pelin koodissa.

Pelissä tietyllä hetkellä ohjelman käytettävissä olevat korttioliot valitaan kokonaisuutena taulukon muuttujan chapter arvon avulla. Taulukosta valitaan oikea kortti näytettäväksi muuttujan index arvon avulla.



Kuva 36: Miten koodin ajaminen näkyy pelaajalle. Kuvankaappaus Iltan jakson alusta pelin prototyypistä.

Kokonaisuudessaan pelin koodin ajaminen toimii niin, että JavaScript-koodia ei ajeta heti verkkosivun latauduttua. Koko pelin toiminnasta vastaavaa funktiota Peliä kutsutaan vasta kun pelaaja painaa aloita peli -nappia.

8.7 Kommentointi koodissa

Koodia on kommentoitu kirjoittamisen ohessa laajalti itse koodauksen helpottamiseksi, sekä myöhempää tarkastelua varten. Kommentoinnista oli merkittävää hyötyä projektissa, etenkin pitkien ohjelmointitaukojen jälkeen. Koko projekti kesti yli vuoden, joten vanhojen koodin osien palauttaminen mieleen helpottui merkittävästi kommentoinnin avulla.

8.8 Äänisuunnittelu

Peli on toteutettu niin, että sitä on mielekästä pelata useita kertoja. Tarina muuttuu ja on erilainen eri pelikerroilla. Peliä on kehitetty yhdessä nuorten kanssa ja siinä on paljon elementtejä, joita nuoret pitivät hyvän pelin ominaisuuksina, kuten äänet ja musiikki, jotka on

tehty tätä peliä varten pitämään yllä mielenkiintoa ja tekemään pelikokemuksesta paremman. Ääni kuuluu, kun uusi kortti kääntyy. Musiikki on luotu sopimaan jokaisen jakson ainutlaatuisen tapahtumapaikkaan ja hahmon toimintaympäristöön. Äänen saa myös kytkettyä pois päältä.

”Kitara on nauhoitettu äänikortin kautta suoraan koneelle, efektit ja vahvistin lisätty Neural DSP:n plugineilla. DAW:ina (Digital Audio Workstation) on käytetty Presonuksen Studio Onea. Kaikki mitä ei ole tehty kitaralla itse, on tehty tietokoneohjelmalla, johon on ostettu VST-plugineja (Virtual Studio Technology), joka mallintaa esim. orkesteri- tai elektronisia soittimia ja rumpuja” (Haikonen 2022). Äänisuunnittelija Taneli Haikosen mukaan keskivertopelaaja katsoo lopetusruutua n. 2-5 sekuntia, eikä lyhyiden teemakappaleiden tekeminen ole muutenkaan luontevaa. Pidempien, n. 1:30-2:30 kestävien teemakappaleiden toteuttaminen sen sijaan on.

8.9 Pelin viimeistely

Pelin testauksessa saatiin palautetta monesta asiasta. Yksi näistä oli graafinen ulkoasu. Tekstit olivat epäselviä. Pelissä oli pitkään melko pieni renderöintiresoluutio. Tämä korjattiin tuplaamalla pelin sisäinen renderöintiresoluutio, jonka avulla teksti saatiin näyttämään terävältä. Parannus oli huomattava ja tekstin luettavuus sekä ulkoasu paranivat merkittävästi.

```
canvas.style.width=size+"px";
canvas.style.height=size+"px";
window.devicePixelRatio=2;

let scale=window.devicePixelRatio;
ctx.scale(scale, scale);
canvas.width = Math.floor(size * scale);
canvas.height = Math.floor(size * scale);
ctx.setTransform(2, 0, 0, 2, 0, 0);
```

Kuva 37: Pelin renderöintiresoluution tuplaus koodissa.

Renderöintiresoluution tuplaamisella ei ollut käytännössä mitään negatiivista vaikutusta pelin suorituskykyyn. Ratkaisu oli itsessään toimiva ja jätettiin peliin sellaisenaan.

8.10 Lisäominaisuudet peliin palautteen pohjalta

Peliä testatessa tuli ilmi tarpeita lisäominaisuuksille. Testausvaiheessa eräällä testaajalla oli käynyt niin, että hän oli klikannut vahingossa kaksi kertaa, tehnyt tämän takia väärän valinnan ja hypännyt kortin yli. Tämä ongelma korjattiin lisäämällä peliin klikkausnopeuden rajoitinfunktio, jota kutsutaan joka kerta kortin vaihtuessa.

```
function reduceClickSpeed(){
  kylla.disabled=true;
  ei.disabled=true;
  valinta.disabled=true;
  setTimeout(activateButtons, 300);
}
```

Kuva 38: Valintanäppäimien lukituksesta vastaava funktio reduceClickSpeed.

Tämä funktio lukitsee pelin valintanäppäimet 300 millisekunnin ajaksi niin, että niitä ei voi painaa sinä aikana. Näin estetään vahinkotuplaklikkaukset kuitenkin nopeasti eteneviä pelaajia hidastamatta.

8.11 Koodin viimeistely

Pelin koodissa oli projektin loppuvaiheessa paljon turhia osia ja epäselviä kohtia. Projektin lähestyessä loppuaan poistettiin koodista turhia osia ja kommentteja. Poistettiin tyhjiä rivejä koodista ja paikoin lisättiin tyhjiä rivejä koodin selkeyttämiseksi. Yhdistettiin liian hajanaisia funktioita yhdeksi ja hajautettiin liian yhdistettyjä funktioita moneksi. Näin parannettiin koodin luettavuutta. Testauksen aikaisia kommentteja myös kuitenkin jätettiin koodiin viimeisimpien ongelmien mahdollista myöhempää ratkaisemista varten. Peli on suunniteltu julkaistavaksi niin, että mikäli ongelmia ilmenee vielä julkaisun jälkeen, voidaan ne ratkaista. Näin tehtiin julkaisun jälkeen bugille, joka ilmeni silloin, kun pelaaja päätyy kuolemaruutuun ja aloittaa pelin alusta. Kuolemaruudun musiikki jäi soimaan uudelleen aloitetun pelin musiikin päälle; tämä virhe korjattiin.

9 Bugien korjaaminen

Pelin ohjelmoimisen edetessä ilmeni monenlaisia teknisiä ongelmia ja bugeja. Näiden ratkoomiseen kului merkittävästi aikaa, mutta lopulta kaikki löydetyt bugit saatiin korjattua tai ongelmat kierrettyä. Peliä koodatessa ongelmia ilmeni mm. kuvien näkyvyyden kanssa, pelin

uudelleenkäynnistyksen kanssa ja lisäksi kortteja vaihdettaessa. Tässä osiossa avataan merkittävien bugien löytymisen ja korjaamisen prosessia.

9.1 Korttolioiden sisältö

Tarinaa koodatessa itse peliin Miron hahmotelmien pohjalta ilmeni ongelmia pelin korttien täyttämässä. Pelin kortit täytettiin yksi kerrallaan manuaalisesti koodiin kirjoittamalla. Tätä tehdessä sattui kirjoitusvirheitä ja pieniä, pelin rikkovia virheitä korttoliota muodostaessa. Nämä täytyi etsiä yksi kerrallaan pelaamalla peliä toistuvasti uudestaan ja etsimällä kohtia, jossa tarina ei toimi kuten pitäisi, tai jossa peli itsessään lakkaa toimimasta. Tämän tyyppisten kirjoitusvirheiden korjaaminen oli melko helppoa, mutta niiden löytäminen vaikeaa. JavaScript toimii niin, että yksikin havaitsematon syntaksivirhe voi rikkoa koko ohjelman. Tämän takia korttoliota täytettäessä täytyi olla erityisen tarkkana. Virheet eivät ilmenneet aina heti, usein niiden ilmeneminen vaati tietyt olosuhteet pelissä, kaikki muut tilanteet pelissä saattoivat toimia. Tästä johtuen testaaminen ja bugien korjaaminen vei todella paljon aikaa.

```
let iltavaihto4=new Dialogi(78, "Kenet tapaavat seuraavaksi?", ["Laine", "Manta", ""], [{"m:12,v:3},{m:0, v:250}], [{"m:12,v:4},{m:0, v:250}],0,0);
```

Kuva 39: Esimerkki korttoliosta sisältöineen, jokainen pelissä näkyvä kortti on korttoliota.

Dialogi-luokan korttoliota sisään asetetaan kuvan 39 mukaisesti järjestyksessä: järjestysluku ilmaiseva numero, (ei itse järjestysluku), joka on ohjelmoijaa varten korttien järjestyslukujen hahmottamisen mahdollistamiseksi, luku itsessään ei tee mitään; kortin tekstisisältö; nappien tekstisisältö; taulukko joka sisältää funktiot ja niiden arvot joita kutsutaan pelaajan painaessa ”kyllä”; toinen samanlainen taulukko mutta vaihtoehdolle ”ei”; seuraavan kortin järjestysluku, jos pelaaja painaa ”kyllä”; seuraavan kortin järjestysluku, jos pelaaja painaa ”ei”.

```
constructor Dialogi(id: any, tarinaTeksti: any, napit: any, kylla: any, ei: any, seuraavakylla: any, seuraavaEi: any, valinta: any, seuraavaValinta: any): Dialogi
```

Kuva 40: Korttoliota osat selitettynä ilman sisältöä.

Korttoliota täyttäminen Mirossa olleella sisällöllä oli merkittävä osa projektiin kulunutta kokonaisaikaa. Työ oli toistuvaa ja tarkkaa. Toistuvien koko koodin läpikäyntien ja pelin jatkuvan testaamisen avulla saatiin koko pelin sisältö toimimaan ilman havaittavia virheitä.

9.2 Pelin uudelleenkäynnistyksen ongelmat

Pelin uudelleenkäynnistystoiminto toimi pitkään niin, että yksinkertaisesti koko verkkosivu ladatain uudestaan pelaa uudelleen -nappia painettaessa. Myöhemmässä vaiheessa loppukesästä-alkusyksystä 2023 peliin koodattiin kunnollinen uudelleenpelaamisfunktio. Koko pelin uudelleenlataaminen toimi sinänsä hyvin, mutta todellisen palvelimen kautta pelattuna tämä aiheutti sen, että kaikki pelin tiedostot pitää ladata uudestaan joka pelikerralla, joka vie dataa ja mahdollisesti aikaa hitaalla internetyhteydellä pelattaessa. Tämän välttämiseksi oli syytä tehdä toimiva uudelleenpelaamisfunktio. Alun perin uudelleenpelaamisfunktiota rakennettaessa syksyllä 2022-kevällä 2023 ilmeni teknisiä haasteita, joiden ratkaiseminen ei ollut ajankäytöllisesti järkevää siinä vaiheessa, jolloin päädyttiin sivun uudelleenlataamiseen perustuvaan tilapäiseen ratkaisuun, joka oli täysin riittävä pelin prototyypivaihetta ajatellen.

Valmiissa versiossa uudelleenpelaamisfunktio eli `restartGame` toimii niin, että se nollaa kaikki pelissä tarvittavat pelilogiikan arvot, pysäyttää musiikin, lataa alkuruudun uudestaan ja poistaa ylimääräiset `EventListener`it pelistä tuplakutsujen välttämiseksi. Tuplakutsuilla tässä yhteydessä tarkoitetaan esimerkiksi kyllä-napin painamisesta pelissä aiheutuvia vaikutuksia. Koodin alkupuolella lisätään kyllä-napille `EventListener`, joka kutsuu funktioita korttoliioon kirjoitettujen vaikutuksien mukaisesti. Mikäli koodin alkupuolella lisättyä `EventListeneriä` ei poistettaisi tässä pelaa uudelleen -funktiossa, kutsuttaisiin kyllä-nappia painettaessa siitä aiheutuvat vaikutukset kahteen tai useampaan kertaan, joka ei itsessään riko koko pelin toimintaa, mutta rikkoo sen järkevän toiminnallisuuden pisteiden kannalta täysin. `restartGame` -funktioon on kirjoitettu myös alkuruudun opastetekstit uudelleen, jotta kokemus alkuruudusta vastaa ensimmäistä pelikertaa uudelleen pelattaessa.

9.3 Animaatiobugi

Pelissä oli pitkään ongelmana suuri tietokoneen resurssien, erityisesti näytönohjaimen käyttö. Laajan testaamisen jälkeen todettiin, että pelin pisteiden lisäämiseen sekä vähentämiseen liittyvät animaatiot, jotka piirtävät pistekategorioiden yläpuolelle plus- ja miinusmerkkejä jäävät pyörimään taustalle, vaikka ne menisivät ulos ruudusta. Tähän rakennettiin ratkaisuksi ajastin, joka pysäyttää animaatiot siinä vaiheessa, kun plus- ja miinusmerkit ovat kadonneet ruudulta. Tämä ajastin vähentää tietokoneen grafiikkasuorittimen käyttöä merkittävästi ja peli toimii huomattavasti pienemmällä tietokoneen resurssienkäytöllä.

9.4 Kuvien näkyvyysongelma

Peliä rakennettaessa oli monessa vaiheessa ongelmana korttien sisälle piirrettyjen kuvien näkyvyys. Kehityksen loppuvaiheessa kuvat toimivat hyvin silloin, kun peliä pelattiin suoraan paikallisista tiedostoista avattuna selaimessa. Kuitenkin todellisessa palvelinympäristössä, tässä tapauksessa Netlifyssa (Netlify 2023) pelattaessa kaikki kuvat eivät latautuneet oikein. Ongelma saatiin osin ratkottua järjestelemällä korttien piirtämisestä vastaavaa funktiota `drawCard` ja muodostamalla tämän ulkopuolelle vielä erillinen `drawInCard` -funktio, joka vastaa kortin sisälle piirretyistä taustakuvista ja hahmokuvista. Ongelman lopulliseksi ratkaisemiseksi käytettiin asynkronisia funktioita, jotka kutsutaan funktion `drawInCard` sisällä vasta kuvien latauduttua.

```
function drawInCard(){//19.9 yhdistetty eri korttiluokkien kuvien piirtäminen tähän.
  function drawActualImage(){
    ctx.shadowColor="black";
    ctx.shadowBlur=15;
    let img=korttiKuvat[index];
    if(img.complete){
      // console.log("img.complete");
      ctx.shadowColor="black";
      ctx.shadowBlur=15;
      if(largeScreenOn==0){
        ctx.drawImage(korttiKuvat[index], x - 125, y - 130, 250, 250);
        drawHahmo();
      }
      else if(largeScreenOn==1){
        ctx.drawImage(korttiKuvat[index], x-150, y - 150, 300, 300);
        drawHahmo();
      }
      drawHahmo();
    }
    else if(!img.complete){
      img.onload=function(){
        ctx.shadowBlur=15; //tämä korjasi varjo-ongelman (varjo ei näkynyt ihan ensimmäisessä kuvassa)
        ctx.shadowColor="black";
      }
      if(largeScreenOn==0){
        ctx.drawImage(korttiKuvat[index], x - 125, y - 130, 250, 250);
        drawHahmo();
        // console.log("hahmoKuvaladattu");
      }
      else if(largeScreenOn==1){
        ctx.drawImage(korttiKuvat[index], x-150, y - 175, 300, 300);
        drawHahmo();
      }
    }
  }
  drawHahmo();
}
```

Kuva 41: Kuvankaappaus funktiosta `drawInCard`, jonka vastuulla on korttien sisään piirretyt kuvat.

Funktiossa `drawInCard` testataan, ovatko kuvat latautuneet ja toimitaan sen mukaisesti. Mikäli kuva on latautunut, ohjelma piirtää sen suoraan ruudulle. Jos se ei ole latautunut, muut pelilogiikan osat suoritetaan välittömästi ja kuva piirretään ruudulle heti sen latauduttua. Tämä toimi loistavasti silloin, kun peliä testattiin paikallisessa ympäristössä. Oikeassa palvelinympäristössä kuvien latautuminen oli silti paikoin hidasta, joten kuvan kohdalle asetettiin ladataan-teksti kuvien latautuessa pelaajan opastamiseksi. Ratkaisu oli lopulta melko yksinkertainen, mutta sen löytämisessä kului paljon aikaa.

Kuvien kanssa ilmeni kehityksen aikana myös sellainen ongelma, jossa kuvan tiedostoa ei löytynyt. Tähän löytyi lopulta yksinkertainen ratkaisu: pelin JavaScript-koodiin oli kirjoitettu kuvien tiedostonimet joissain kohdissa väärin. Nämä korjaamalla kuvat näkyivät oikein.

```

kuvat[31] = { kuva: "images/ymparistot/laine/jarviranta.png", tkuva: "images/ymparistot/laine/jarvikesa.png", hkuva: "images/hahmot/laine_vihainen_pieni.png" };
kuvat[32] = { kuva: "images/ymparistot/laine/jarviranta.png", tkuva: "images/ymparistot/laine/jarvikesa.png", hkuva: "images/hahmot/laine_vihainen_pieni.png" };
kuvat[33] = { kuva: "images/ymparistot/laine/jarviranta.png", tkuva: "images/ymparistot/laine/jarvikesa.png", hkuva: "images/hahmot/laine_profiili_iloinen.png" };
kuvat[34] = { kuva: "images/ymparistot/laine/jarviranta.png", tkuva: "images/ymparistot/laine/jarvikesa.png", hkuva: "" };
kuvat[35] = { kuva: "images/ymparistot/laine/jarviranta.png", tkuva: "images/ymparistot/laine/jarvikesa.png", hkuva: "" };

```

Kuva 42: Esimerkki kuvien korttikohtaisesta määrittämisestä.

Kuvien korttikohtaisessa määrittelyssä piti olla todella tarkkana kuvan 42 mukaisesti. Korttien määrän kasvaessa kirjoitusvirheiden riski kasvoi, eikä Visual Studio Code antanut mitään virheilmoitusta näistä, sillä kaikki oli kirjoitettu teknisesti oikein, ainoastaan kuvan tiedostonimeen viittaavissa teksteissä oli virheitä.

```

for (let i = 0; i < kuvat.length; i++) {
    korttiKuvat[i] = new Image();
    taustaKuvat[i] = new Image();
    hahmoKuvat[i] = new Image();

    korttiKuvat[i].src = kuvat[i].kuva;
    taustaKuvat[i].src = kuvat[i].tkuva;
    hahmoKuvat[i].src = kuvat[i].hkuva;
}
}

```

Kuva 43: Kuvaolioiden muodostaminen kuvalistasta ja näiden asettaminen erillisiin korttikuva, taustakuva ja hahmokuvalistoihin.

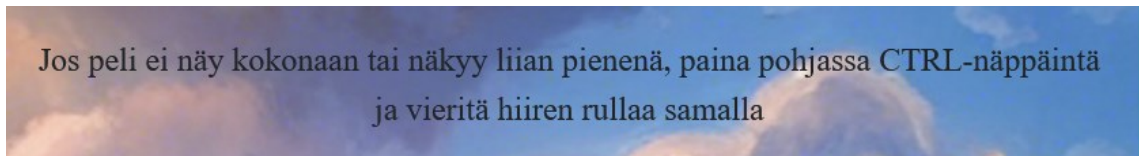
Kuvien asettaminen kuvalistaan ja kuvan 43 mukaisesti listan läpi iteroimalla for-silmukan avulla saatiin kuvakortit asetettua korttiolioihin dynaamisesti ja renderöityä ne pelaajalle näkyviksi kätevästi kuvaolioista. Jokaisesta korttioliosta löytyy tämän silmukan avulla asetettu muuttuja kortin sisäiselle taustakuvulle (koodissa korttiKuva), kortin takana näkyvälle taustakuvulle (taustaKuva) ja hahmokuvalle (hahmoKuva), joka sisälsi kyseisen kortin hahmokuvan. Tämä muuttuja sisälsi peliprojektin kansiossa sijaitsevan kuvan tiedostoon johtavan tiedostopolun. Kuvia käytettiin tarpeen mukaan, kaikista korttiolioista löytyy kaikki kuvaluokat, mutta silloin kun hahmokuvaa ei tarvittu, jätettiin kyseisen muuttujan arvoksi tyhjä merkkijono.

9.5 Konsolin käyttö bugien korjaamisessa

Koko peliprojektin ajan käytettiin paljon JavaScriptin sisäänrakennettua console.log-metodia. Tämän metodin avulla voidaan määritellä koodiin tilanteita, joissa peliä pelattaessa näkyy ohjelmoijan määrittelemä teksti tai esimerkiksi tietyn muuttujan arvo selaimen kehittäjätyökaluissa. Kehittäjätyökalunäkymä avautuu yleisimmissä selaimissa Windowsissa painamalla F12-näppäintä. Konsoli-ilmoitusten avulla voitiin päätellä, mikä kohta koodissa ei toimi. Peliä koodatessa ongelmia löytyi console.login avulla odottamattomista paikoista, console.log oli korvaamaton työkalu ohjelmoinnin onnistumisen kannalta.

9.6 Ongelmat pelin kuvaruudun skaalautumiseen liittyen

Peliä ohjelmoitaessa peliä testattiin eri kokoisilla näytöillä ja ikkunoiden ko'oilta. Ohjelmoinnissa käytettiin kuitenkin pääosin suurikokoisia pöytätietokonenäyttöjä (3840x2160 pikseliä, 2560x1440 pikseliä ja 1920x1080 pikseliä) ja suuriresoluutioista kannettavan tietokoneen näyttöä (2560x1600 pikseliä). Peliä ohjelmoitaessa kokeiltiin suurentaa peliruutua merkittävästi oletusarvoa isommaksi pienemmän ruudun jäljittelemiseksi. Pelin käyttäjätestausvaiheessa tuli ilmi, että monella pelaajalla oli ohjelmoinnissa käytettyjä näyttöjä merkittävästi pienempi näyttö, jossa peli ei näkynyt oikein ohjelmointivaiheen suurennetulla kuvalla toteutetuista testauksista huolimatta. Ongelmana oli se, että peli oli edelleen liian suuri näytöllä pystysuunnassa. Tähän tehtiin ratkaisuksi funktio measureWidth, joka mittaa näytön koon joka kerta kun ikkunan kokoa muutetaan, pelin ensimmäinen latauskerta mukaan lukien. Tieto ruudun koosta välitetään tältä funktioltä eteenpäin pelin itsensä piirtäville funktioille muuttujan largeScreenOn muodossa, joka näistä piirtämisfunktioista käsin vaikuttaa peliin niin, että largeScreenOnin arvon ollessa 1 piirretään suurikokoinen peli ja arvon ollessa 0 piirretään pienikokoinen peli. Testaajien mukaan tämä muutos korjasi ongelman.



Kuva 44: Alkuruudun opasteteksti selaimen zoomaustason muuttamiseksi.

Opasteteksti tehtiin peliin ennen pelikuvan skaalauksen onnistumista. Se jätettiin kuitenkin peliin mahdollisten äärimmäisten tapausten eli todella suurten tai todella pienten näyttöjen käyttämisen selkeyttämiseksi.

10 Yhteenveto ja arviointi

Opiskelijoiden idea Munuais- ja Maksaliiton uudelle digitaaliselle palvelutuotteelle syntyi keväällä 2022 tehdyn Palveluiden käyttäjäkeskeisen suunnitteluun liittyvän kyselytutkimuksen ja sitä seuranneen pohdinnan yhteydessä. Projektin alkaessa varsinaisesti elokuussa 2022 peli ei ollut vielä saanut muotoaan. Ilmassa oli innostusta ja tekemisen meininkiä, kokouksissa keskusteltiin ideoista peliin liittyen sekä saatiin liiton ja sidosryhmien asiantuntijoilta ohjeita pelin sisällön suhteen. Toimittajatiimissä sisällöntuottaminen Miro- alustalla toimi hyvin, mutta se olisi voinut toimia vieläkin tehokkaammin ja paremmin, sillä se on todella hyvä työkalu yhteiskehittämiseen. Myös asiakkaalle annettiin käyttöoikeudet Miroon. Myös pelin ohjelmoinnissa käytetty GitHub mahdollistaa loistavasti yhteiskehittämisen, mutta kaikki opiskelijat eivät tätä mahdollisuutta hyödyntäneet.

Kotimaisia oppimispelejä on kyllä saatavilla, kuten taustatutkimuksen yhteydessä havaittiin. Osoittautui myös, että tutkimukseen perustuvien oppimispeleiden kehittämiseksi voisi olla Suomessa tilaisuus, jos vaikkapa startup-yrittäjyys kiinnostaisi tämän pelin tekijöitä myöhemmin. Käyttäjäkeskeinen suunnittelu on laadukkaan palvelu- sekä pelinkehittämisen edellytys. Esitutkimuksessa tulleita havaintoja hyödynnettiin pelinkehityksessä ja Fantasia-ankesälomapeliin päätyi samoja ominaisuuksia kuin vertailuissa oppimis- sekä muissa peleissä. Teknisiltä ominaisuuksiltaan hyvin toimiva valintapeli, näyttävä ja visuaalinen toteutus sekä dialogi hahmojen sekä pelaajan välillä uniikissa tarinassa yhdistettynä toimivaan äänimaailmaan olivat käytettyjen palvelumuotoilun työkalujen mukaan hyviä, kohderyhmälle tarkoitettua uuden palvelutuotteen ominaisuuksia. Asiakkaan tarpeiden ja toiveiden mukainen uusi digitaalinen palvelu tulee tarjoamaan käyttäjilleen mukavan peli- ja oppimiskokemuksen. Hauskalla sekä nuorille sopivalla tavalla kerrotun tarinan tarkoituksena on antaa ja jakaa tietoisuutta paremmista sekä terveyttä tukevista valinnoista.

Paljon pelikokemusta sekä myös omaa sairaushistoriaa munuais- ja maksasairauksista omaavat nuoret antoivat arvokasta tietoa opiskelijatiimille, jolla oli myös runsaasti omaa kokemusta ja

kiinnostusta erilaisista digi-, konsoli- sekä roolipeleistä. Yhteistyö toimi hyvin ja tarinallinen valintapeli alkoi hahmottumaan fantasiamaailmaan, johon graafikko/dramaturgi loi kerto-
musta hahmoineen. Myös opiskelijat osallistuivat tarinan tekemiseen. Pelin maailma on tai-
anomainen paikka, jossa seikkailevat upeat piirroshahmot ovat haastamassa pelaajan teke-
mään hyviä valintoja ajatellen terveyttä ja hyvinvointia. Audiovisuaalisessa pelissä on tarinaa,
tunnelmia ja tapahtumia eteenpäin kuljettava äänimaailma sekä musiikki, mikä on mahdol-
lista myös hiljentää.

Luovassa, taiteellisessa ja teknisessä projektissa työskenneltiin tiimissä sekä itsenäisesti. Ske-
naariot tässä pelissä jäivät hieman liian monimerkityksellisiksi, tämä voi paikoittain tuntua
pelaajasta hämmentävältä. Valitettavasti tässä kohtaa näkyy tarinankertojan kokemattomuus
ja kyvyttömyys tuottaa selkeästi erilaisia skenaarioita kuten vaikkapa Roihun kappaleessa,
jossa teemat hieman sekoittuvat, eikä korttien järjestystä voinut enää pelinkehityksen loppu-
vaiheessa muuttaa niin että skenaariot olisivat keskittyneet selkeämmin tupakointiin, pelaa-
miseen ja rentouttavan hengittämisen tärkeyteen niin, että myös vaikutukset kappaleen lope-
tuksiin olisivat olleet erilaisia. Esi- ja taustatutkimuksen yhteydessä olisi voitu kysyä nuorilta
enemmän heidän vapaa- ajan viettotavoistaan sekä etenkin siitä, minkälaisissa ympäristöissä
he viettävät kesälomaansa. Nyt tämä jäi hieman arvailun varaan, koska tarinankerronnan si-
jaan keskityttiin enemmän pelin ominaisuuksiin.

Tekijätiimi on kuitenkin työssään ottanut huomioon asiakkaan tarpeita sisällön suhteen ja toi-
minut annettujen suuntaviivojen mukaisesti, vaikka pelistä tuli hieman syvällisempi kuin asi-
akkaan toive ehkä oli. Kaikki peliin tuotettu asiasisältö on peräisin luotettujen ja tunnettujen
yhteiskunnallisten järjestöjen, virastojen sekä muiden toimijoiden tutkimasta ja tuottamasta
tiedosta sekä lähteistä. Asiakas ja sidosryhmät ovat antaneet tiimille vapauden luoda palvelu-
tuotteen opiskelijoihin luottaen ja koko ajan tukien sekä ohjaten ja myös korostaen pelin tär-
keää terveydellistä opetusisältöä. Kaikki kommunikaatio on ollut rohkaisevaa ja positiivista,
tavoite on ollut yhteinen ja selkeä. Tämä lopputyö on 15 opintopisteen arvoinen kokonaisuus.
Se on sisältänyt paljon työtä, jota on tehty kaikkina vuorokaudenaikoina motivoituneina, si-
toutuneina ja innostuneina. Ohjausta työhön on asiakkaan lisäksi tarvittaessa saatu myös Lau-
rean tietojenkäsittelyn lehtorilta.

Pelin julkaisutilaisuus oli miellyttävä ja lämminhenkinen julkaisutapahtuma, joka antoi myös
mahdollisuuden työryhmälle tavata toisiaan vihdoin kasvokkain suurimmaksi osaksi etätyös-
kennellyn vuoden jälkeen. Pelistä tehtiin myös opettajan opas, jonka Munuais- ja maksaliitto
toteutti pelin julkaisun jälkeen omalla aikataulullaan ilman opiskelijoita. Opettajan ohjeita ja
oppilaskyselyn tuloksia ei esitellä tässä raportissa. Opiskelijat olivat vielä myös mukana pohti-
massa kysymyksiä oppitunneilla pelattavan pelin käyttäjille, kyselyn avulla tutkittaisiin pelistä
saatuja kokemuksia ja kerättäisiin palautetta pelistä vielä aktiivisen pelinkehityksen jälkeen-
kin. Projektin alkuperäinen aikataulu oli pelin lopulliseen kokoon ja työmäärään nähden aivan

liian suppea tai vaihtoehtoisesti pelistä tuli työmäärältään laajempi kuin aluksi oli kuviteltu. Toisaalta myös kausiluonteiset taudit työskentelystä venyttivät hieman projektin aikataulua.

Lähteet

Painetut

Bergström S., Leppänen A. 2016. Yrityksen asiakasmarkkinointi. Edita.

Koivisto M., Säynäjäkangas J., Forsberg S. 2019. Palvelumuotoilun bisneskirja. Alma Talent.

Krokkfors L., Kangas M., Kopisto K. (toim.) 2014. Oppiminen pelissä- Pelit, pelillisuus ja leikkisyys opetuksessa. Hansaprint Oy.

Raatikainen L. 2008. Asiakas, tuote ja markkinat. Edita.

Ruuska K. Pidä projekti hallinnassa- suunnittelu, menetelmät, vuorovaikutus. 2012. Hansaprint Oy.

Tuominen K., Niva M. 2022. Benchmarking käytännössä- kehittämisopas. CMP.

Sähköiset

American National Standard Flowchart Symbols and Their Usage in Information Processing, X3.5-1970.1970. ANSI: New York: American National Standards Institute. Viitattu 7.11.2023.

https://webstore.ansi.org/preview-pages/iso/preview_iso+5807-1985.pdf?_gl=1*52s8qs*_gcl_au*NDI2NDgwMjU2LjE2OTkzNDA2NzA.

Artlex 2022. Viitattu 22.12.2022. <https://www.artlex.com/blog/character-concept-art-ideas/>

Borislav Hadzhiev. Bobbyhadz Blog. Viitattu 3.11.2023. <https://bobbyhadz.com/>

Chapin, N. 1971. Flowcharts. New York: Petrocelli Books.

Dude 2019. Nettisivujen skaalautuminen. Viitattu 28.6.2023. <https://dudeunlimited.fi/wp-content/uploads/2019/03/nettisivujenskaalautuminen.pdf>

eLuotsi. Responsiivisuus. Viitattu 28.6.2023. <https://www.eluotsi.fi/responsiivisuus/>

Fullerton, T. 2008. GAMEDESIGN WORKSHOP A Playcentric Approach to Creating Innovative Games.

Geeks For Geeks. Viitattu 3.11.2023. <https://www.geeksforgeeks.org/>

Hamari, J. 2017. Viitattu 14.10.2022. <https://yle.fi/uutiset/3-9600219>

Hämäläinen P., Takatalo J. 2017. Duodecim. Viitattu 9.11.2022. <https://www.duodecim-lehti.fi/duo14046>

Laatumäärittely. 2021. Opetushallitus. Viitattu 9.11.2022. <https://uudetlukutaidot.fi/wp-content/uploads/2021/12/Pedagogisesti-laadukas-digitaalinen-ymparisto.pdf>

Markkinointiliitto. Viitattu 15.4.2023. <https://www.markkinointiliitto.fi/menneet-tapahtumat/mark-koulutus-insightista-konseptiksi/>

MDN Web Docs A. Mozilla. Viitattu 4.11.2023. https://developer.mozilla.org/en-US/docs/Learn/JavaScript/Objects/Object-oriented_programming

MDN Web Docs B. Mozilla. Viitattu 4.11.2023. <https://developer.mozilla.org/en-US/docs/Learn/JavaScript/Objects>

MDN Web Docs C. Mozilla. Viitattu 9.11.2023. <https://developer.mozilla.org/en-US/docs/Web/JavaScript>

Munuais- ja maksaliitto Viitattu 9.11.2022. <https://www.muma.fi/>

Mannerheimin Lastensuojeluliitto Viitattu 12.12.2022. <https://www.mll.fi/vanhemmille/vinkkeja-lapsiperheen-arkeen/nuori-kiinnostuu-alkoholista/>

Netlify. Viitattu 9.11.2023. <https://www.netlify.com/>

Northmanvr. Viitattu 9.11.2023. <https://northmanvr.com/fi/palvelut/>

Palma. Viitattu 13.6.2023. <http://palma.fi/palma-malli/tuotteistaminen/?tab=2>

Pelikasvatus. Pelikasvattajan käsikirja. Viitattu 27.9.2022. <https://www.pelikasvatus.fi/pelikasvattajankasikirja.pdf>

Psyongames. Viitattu 31.10.2023 https://psyongames.com/?fbclid=IwAR0SewvdwO5RXYwS-DsMQWw-d_mUJ9rNK2limDh4cDhLCF-QMRC56wleAjKM

Toim. Rahmel P. 2019, 2021. TARU oppimateriaalipankki- tarinallisuus pedagogiikassa. Metropolia.

Rubin A. tulevaisuus.fi. Viitattu 6.6.2023. <https://tulevaisuus.fi/metodit/skenaarioajatelu-tulevaisuudentutkimuksessa/skenaarion-kasitteesta/>

Storyboard That. Viitattu 12.9.2022. <https://www.storyboardthat.com/fi/matkan-kartta-k%C3%A4ytt%C3%A4j%C3%A4yst%C3%A4v%C3%A4llinen-empatiaa-kartalla>

SurveyMonkey. Viitattu 3.4.2023. <https://fi.surveymonkey.com/mp/comparing-closed-ended-and-open-ended-questions/>

Stability AI. Viitattu 3.11.2023. <https://stability.ai/use-policy>

Stack Overflow. Viitattu 3.11.2023. <https://stackoverflow.com/>

Tekijänoikeus.fi Viitattu 25.4.2023. <https://tekijanoikeus.fi/tekijanoikeus/>

Tarinakone. Viitattu 22.1.2023. <https://tarinakone.fi/tarinallistaminen/>

TV Tropes. Viitattu 9.11.2023. <https://tvtropes.org/pmwiki/pmwiki.php/Main/BigLippedAlligatorMoment>

W3Schools. Viitattu 3.11.2023. <https://www.w3schools.com/>

Julkaisemattomat

Ehyt. 2022. Työryhmän kokouksessa.

Haikonen T. 2022. Pyydetty lausunto Fantasiakesälomapelin äänisuunnittelusta WhatsAppissa.

Munuais- ja maksaliitto. 2023. Kokousmateriaalista.

Vaikkola L. 2022. Työryhmän kokouksessa.

Kuviot

Kuvio 1: Munuais- ja maksaliiton digitaaliset palvelut peliprojektin alussa	10
Kuvio 2: Projektin suhde ympäristöön	11
Kuvio 3: Havainnot ja johtopäätökset	16
Kuvio 4: Empatiakartta.....	18
Kuvio 5: Toimeksiantajan ja yhteistyökumppaneiden sanoja pelin sisällöstä.	19
Kuvio 6: Tärkeinä ja hyvinä pidettyjä pelin ominaisuuksia.....	23
Kuvio 7: Huonoina pelin ominaisuuksina mainitut.....	24
Kuvio 8: Ideointia Mirossa. Peli-ideat ovat omissa kategorioissaan eri väreillä merkattuina.	25
Kuvio 9: Pelin suunniteltu eteneminen kaavion kautta Mirossa.	26
Kuvio 10: 1. Prototyypin tarina Mirossa.....	28
Kuvio 12: Ote Roihun kappaleesta Mirossa skenaarioineen.....	52
Kuvio 13: Iltan kappaleen skenaarioita Mirossa.	53

Kuvat

Kuva 1: Pelin aloitusnäkyvä	14
Kuva 2: Arjen turvallisuuden STOP-peli.	17
Kuva 3: Pisteet pelin prototyypissä.....	21
Kuva 4: Roihun kappaleen avustaja ja sen kertoma teksti yhdessä tarinan kohdassa.	27
Kuva 5: Peliprototyypin tiedostorakenne ja index.js-tiedoston koodia Visual Studio Codessa 22.4.2023.....	29
Kuva 6: Peliprojektin rakenne ja koodia 6.9.2023. Projektin edetessä koodi muuttui paikoin merkittävästi, mutta kansiorakenne pelin tiedostoille pysyi samana sen muodostuttua.	30
Kuva 7: Korttiluokan Kortti toteutus pelissä.....	32
Kuva 8: Korttiluokka Dialogi, käytetyin korttiluokka pelissä. Muut korttiluokat ovat hyvin samanlaisia ja ovat käytännössä variaatioita tästä korttiluokasta.....	33
Kuva 9: Loppujakson muuttujia.	34
Kuva 10: Funktio changeStats.	35
Kuva 11: Funktioiden kutsuminen olioon määritellyillä arvoilla.....	36
Kuva 12: Taulukko functionArr.	37
Kuva 13: Kuvankaappaus Draw-funktion sisällöstä.	38
Kuva 14: Osio drawCard-funktiosta. Tämä osio aktivoituu, kun pelissä olevan kortin luokka on Dialogi.	39
Kuva 15: Kuvankaappaus drawInCard-funktiosta	40
Kuva 16: Kuvankaappaus funktiosta drawAnimation. Funktio jatkuu hyvin samannäköisenä pitkään. Jokainen pistekategoria on animoitu omalla funktiollaan pisteiden saannin ja menettämisen osalta.	41
Kuva 17: Kuvankaappaus funktiosta changeKuvat.....	42

Kuva 18: Kuvankaappaus funktion <code>changeChapter</code> alusta. Tämä on rivimäärältään pisin funktio koodissa.	43
Kuva 19: Kuvankaappaus Roihun jakson sisällön alkuosasta funktion <code>changeChapter</code> sisällä. Kaikki pelin pelaajalle näkyvä tekstisisältö on kirjoitettu korttiolioihin.	43
Kuva 20: Kuvankaappaus funktiosta <code>die</code>	44
Kuva 21: Kuolemaruutu.	44
Kuva 22: Kuvankaappaus funktiosta <code>showDeathScreen</code> , joka piirtää kuolemaruudun.	45
Kuva 23: Pelin loppuruutu. Pisteluokkia hahmottelevat kuvat ja saatetekstit pelaajan suoriutumisen mukaan kategorioittain.....	46
Kuva 24: Kuvankaappaus funktiosta <code>drawResultScreen</code> , joka piirtää loppuruudun.....	47
Kuva 25: Kuvankaappaus funktiosta <code>restartGame</code>	48
Kuva 26: Prototyypin tila 29.8.2023.	49
Kuva 27: Lisätty osa koodissa.	50
Kuva 28: Manta, Roihu- ja Iltta -hahmot ja niiden ilmeet, joita käytettiin tunnetilojen ilmaisuun pelissä.	55
Kuva 29: Laine-hahmo ensimmäisessä konseptissa.....	56
Kuva 30: Kuva madoista Stable Diffusionilla tehtynä.	57
Kuva 31: Vastaaajien avoin palaute pelistä. Tällä tavoin saatiin osviittaa suoraan nuorilta. ...	59
Kuva 32: Kehitys- ja muokkausideoita testaaajilta.	59
Kuva 33: Lisää pelitestauksen kommentteja	60
Kuva 34: Tuotekortti.....	61
Kuva 35: Jakson aktiivisten korttiolioiden määrittely pelin koodissa.	63
Kuva 36: Miten koodin ajaminen näkyy pelaajalle. Kuvankaappaus Iltan jakson alusta pelin prototyypistä.	64
Kuva 37: Pelin renderöintiresoluution tuplaus koodissa.	65
Kuva 38: Valintanäppäimien lukituksesta vastaava funktio <code>reduceClickSpeed</code>	66
Kuva 39: Esimerkki korttioliosta sisältöineen, jokainen pelissä näkyvä kortti on korttiolio....	67
Kuva 40: Korttiolion osat selitettynä ilman sisältöä.	67
Kuva 41: Kuvankaappaus funktiosta <code>drawInCard</code> , jonka vastuulla on korttien sisään piirretyt kuvat.	69
Kuva 42: Esimerkki kuvien korttikohtaisesta määrittämisestä.....	70
Kuva 43: Kuvaoloiden muodostaminen kuvalistasta ja näiden asettaminen erillisiin korttikuva, taustakuva ja hahmokuvalistoihin.	70
Kuva 44: Alkuruudun opasteteksti selaimen zoomaustason muuttamiseksi.	72

Taulukot

Taulukko 1: Alustava aikataulu peliprojektille (Toteutunut aikataulu löytyy tämän raportin kappaleesta 8.4).....	15
Taulukko 2: Fantasiakesälomapelin Palvelupolku.	20
Taulukko 3: Toteutunut aikataulu	58

Liitteet

Liite 1: Empatiakartan kysymykset	81
Liite 2: Nuorten ideointipajan kysymykset.....	82
Liite 3: Digipelin sisällöt	84
Liite 4: Hahmokonseptin tekijänoikeudet.....	86
Liite 5: Testauskysymykset.....	88

Mikä sua liikuttaa? Mitä sä oikeasti ajattelet ja tunnet? Mikä sulle on tärkeää? Mistä sä haaveilet ja unelmoit?

Miltähän sun ympäristö mahtaa näyttää ja minkälaisia sun ystävät on? Minkälaiset tuotteet ja palvelut on osa sun elämäsi? Minkälaisia ongelmia sä kohtaat?

Miten sä koet sun ympäristön? Vaikuttaako se suhun? Minkälainen vaikutus sun ystävillä on suhun?

Minkälainen sun asenne on ja minkälainen tapa sulla on yleensä käyttäytyä? Ootko sä tosi sosiaalinen vai tykkäätkö olla enemmän itseksesi?

Onko sulla jotain esteitä elämässä? Mikä sua vois turhauttaa kaikista eniten?

Mistä sä tuut tosi iloiseksi? Mikä on sun mielestä menestystä? Mitä sä haluat tavoitella ja saavuttaa?

Liite 2: Nuorten ideointipajan kysymykset

7.10.2022 klo 14-15.30

-Mitä pelejä pelaatte vapaa-ajalla ja millä laitteilla?

-Minkälaista peliä itse pelaisitte?

-Mitä pelissä pitäisi olla?

-Minkälaista peliä ette pelaisi, mitä pelissä ei saa olla?

-Minkälainen peli olisi kiinnostava?

Vastaukset annettiin sekä piirtämällä että kertomalla ja keskustelemalla. Nuoret pohtivat kysymyksiä ryhmissä ja yksilöittäin.

Liite 3: Digipelin sisällöt

Digipelin sisältöjä opiskelijoille

Keskeistä: olisi hyvä olla tietoa siitä, mitkä ovat maksan tehtävät. Tällöin kiellot yms. avautuvat paremmin. Ei liian syvällisesti, mutta kun nuori ymmärtää eri elinten tehtävät, niin omaa toimintaa on myös helpompi muuttaa. Miksi pidän huolta maksastani. Peli voisi alkaa tällä. Voisiko olla esim. viimeisten koulupäivien kokeessa? Peli voisi alkaa esim. terveystiedon kokeella maksan tehtävistä.

Digipeli sijoittuu kesäloman aikaan. Tekemiset jotenkin kytkettävä siihen kesäloma aikaan. Peliin voidaan nostaa muutamat sisällöt keskeisiksi.

Alla olevista asioista ainakin virushepatiitit, alkoholi ja terveelliset elämäntavat olisivat tärkeitä saada mukaan.

Infoa tietokortteihin löytyy esim. Mahtimaksa esitteestä, muista liiton ja Irti huumeista esitteistä. Finellin sivuilla hyvät tiedot esim. kaikkien ruokien ravintoainesisällöistä. Tarkistetut faktatiedot tietokortteihin löytyy näistä teille jo annetuista materiaaleista.

Virushepatiitit - mitä riskejä on esim. ulkomaan matkoilla: ei voi juoda puhdistamatonta vettä (A-hep), miten suojauduin B-hepatiitilta (rokote olemassa A ja B - viruksiin). C-hepatiitti - tarttuu veren välityksellä esim. tatuoinnit, lävistyksset. huumeiden käyttäjien keskuudessa yhteisten neulojen käyttö on riski C-hepatiitille! Älä ota näitä ulkomailla! C-hep. voi tarttua myös suojaamattoman seksin välityksellä, kondomi, ehkäisy. A ja B- hepatiittiin on rokoite; tämä olisi todella hyvää olla olemassa, varsinkin jos menee ulkomaille. C-hep. vastaan ei ole rokotetta. A-hepatiitti tarttuu saastuneesta vedestä ja ruoasta, esimerkiksi huonosti kypsennetyistä äyriäisistä. Kädet on hyvä pestä aina WC:ssä käynnin jälkeen. Ulkomaanmatkoilla kannattaa juoda pullotettua vettä ja välttää jääkuutioita. Ulkomaisia pakastemarjoja tulisi kuumentaa ennen käyttöä 2 minuuttia 90-asteessa.

Mitä on silloin tehtävä, jos on saanut tartunnan? Tämä on myös tärkeä saada peliin. Jotakin oireista myös tähän kohtaan. Milloin on mentävä lääkäriin ja testeihin. Tähän voisi myös laittaa valinnan.

Ylipaino, jokin toinen nimi olisi parempi, vaikka sopiva paino? Miten energiani riittää koko päiväksi? **Keskitytään elintapoihin.** Mikä on kesällä janojuomasi? Vettä suosittava! Mistä saan energiaa? Kesällä ei säännöllistä kouluruokailua. Millainen on esim. hyvä lounas? Tämä voisi olla pelissä. Karkeista ja juomista voisi olla esim. tietoa/arvuuttelu siitä, miten paljon esim. sokeria on limsassa, karkeissa, energiajuomissa? Fruktosilla makeutetut energiajuomat

lisäävät maksan rasvoittumista. Jos esim. syön sipsejä, niin kuinka suuren pussin valitsen. Lyhyitä tietoisuuksia näistä.

Tupakointi, huumeet, nuuska, myrkyt, lääkkeet tai yleisemmin maksalle haitalliset aineet

- mitä ajattelet

näiden tekevän kehollesi, valitsetko käyttäväsi näitä. Nämä kaikki poistuvat maksan kautta ja siksi voisi näistäkin olla jotakin valintatehtäviä. Jos sairastut, otatko lääkettä vai jotakin luontaistuotetta? On tärkeä noudattaa lääkkeiden annosteluohjeita, jos sairaana käytät lääkkeitä. Yliannostus voi vahingoittaa maksaa. Jos käytät luontaistuotteita, kannattaa kertoa lääkärille.

Alkoholi - kokeilut, mitä pitää tehdä, jos kaveri on liian humalassa ja juonut esim. pullon viinaa? Miten käyttäytyminen muuttuu humalassa, riskikäyttäytyminen - ei uimaan/soutelemaan. Palovammat, tapaturmat. Miten hoidan ”krapulaani” - ei oteta uutta ryyppyä tai lääkkeitä vaan juodaan esimerkiksi riittävästi nestettä. Kun käytät paljon alkoholia, mitä elimistössä tapahtuu? Maksa ei ehdi suorittaa kehossa olevaa alkoholia, joten alkoholi vaikuttaa keskushermostaan ja käyttäytyminen muuttuu esim. tapaturmat. Tästä voisi olla valinta ja info-kortti. Mitä tapahtuu alkoholimyrkytyksessä? EHYT:lta saa varmasti tähän apua. Riippuvuus alkoholiin ja muihin aineisiin? Miten kehittyy kehossa ja miten vaikuttaa elämään? Riippuvuus kehittyy osalle käyttäjistä ei suinkaan kaikille.

Liikunta (kävelenkö, vai otanko sähköpotkulaudan, lähdenkö kavereiden kanssa uimaan vai jäänkö kotiin pelaamaan pleikkaa) Jotain jaksamisesta, mielekäs tekeminen, riittävä uni. Kesälomallakin pitää nukkua! Tähän jokin valintatehtävä.

Stressi - miten palaudun koulustressistä, kouluvuodesta, voisi olla siinä alussa jokin valintatehtävä

(myrkkysienet - ei ehkä tarvita tässä pelissä)

Pelissä oli myös kesätyöt - mitä syöt ruokatauolla, on jo pelissä. Tämä on hyvä.

Lomamatkat kavereiden kanssa, mitä riskejä siellä.

Liite 4: Hahmokonseptin tekijänoikeudet

Kuvien tekijänoikeusasioista Moikka! Tässä hieman asiaa kuvitusten tekijänoikeuksista. Kyseessä ei ole niinkään jotain yhtä, virallista tapaa listata näitä asioita, koska jokainen sopimus on erilainen. Kuitenkin, halusin lähettää kirjallisen muistion näistä, jotta kuvien käytössä ei ole epäselvyyksiä! Tekstiä on paljon, mutta älkää pelästykö sitä. Asiat ovat hyvin yksinkertaisia.

Olen kuvittanut teille yhteensä 4-hahmoa + 3 eläinapuria. Kaikista 4:stä hahmosta on pari erilaista versiota, kuten pienemmät ja kroatut versiot. Tämän lisäksi kaikille neljälle hahmolle on piirretty 5 erilaista ilmettä. Kuvat ovat PNG-muodossa.

- Kuvittajana myyn teille kuvatiedostot. En siis myy oikeuksia "hahmoihin" ja heidän konseptihinsa. Tämä tarkoittaa siis lyhkäisyydessään sitä, että itse tiedostot ovat myyty tuote. Muma ei siis osta oikeuksia hahmoihin, heidän nimiinsä tai tarinoihin, eli esim. minä kuvittajana saan yhä käyttää, piirtää ja suunnitella hahmoja omissa projekteissani. (Jos piirrän ja jaan uusia kuvia hahmoista esim. someen, niitä ei tarvitse yhdistää peliin. Kuvittajana en tietenkään aio myydä täysin samanlaisia hahmoja muille mahdollisille asiakkaille, koska hahmot on luotu nimenomaan juuri tähän kyseiseen projektiin.) Saan myös lisätä kuvat someeni ja portfoliooni, mutta minä, tai te ette saa myydä niitä eteenpäin tai pyytää samoista kuvista rahaa muilta asiakkailta.

Esim. Jos asiakas ostaisi oikeudet hahmokonsepteihin ja niistä luotaisiin brändi, en minä eikä kukaan muukaan saisi luoda niistä rahallisia projekteja jatkossa, ellen olisi kuvittajana mukana sopimuksessa.

Kuitenkin: tämä ei tarkoita, etteikö asiakas (siis te) voisi jatkossa tilata esim. lisää kuvia tai tuotteita liittyen juuri kyseiseen aiheeseen tai hahmoihin. Samoista hahmoista voi hyvin luoda lisää, niitä ei tarvitse jäädyttää. Esim. jos asiakas toivoo jo olemassa oleville kuville lisää oikeuksia (kuten niiden käyttöä muualla kuin nettisivulla tai pelissä) asiasta voidaan hyvin sopia.

- Kuvia saa käyttää vain sovittuun projektiin, tässä tapauksessa siis nettipeliin. Lopullisia kuvia saa käyttää myös pelin nettisivulla sekä opiskelijoiden raportissa, credittien kanssa.

- Vain lopullisia valmiita kuvia saa käyttää. Hahmokonsepteja ja hahmojen suunnitteluvaiheessa jaettuja kuvia ei saa jakaa, ne on tarkoitettu vain suunnitteluvaiheessa asiakkaan nähtäville ja arvioitavaksi.

- Kuvia ei saa muokata. Tähän sisältyy esimerkiksi kroppaaminen/rajaaminen, värien muokkaaminen tai mikä tahansa muu joka muuttaa alkuperäistä tuotetta. Kuitenkin, jos jatkossa kyseiselle muokkaukselle tulisi tarvetta, tästä voidaan sopia erikseen minun kanssani.

Tarinasta

Minä ja opiskelijat olemme kaikki antaneet osamme pelin tarinaan. Muma voi keskustella tarinoiden käyttöoikeuksista opiskelijoiden kanssa tarvittaessa (ei välttämätöntä). Koska tarinan teosta ei omalta puoleltani sovittu alussa työtehtävänä ja palkan maksussa, oli kyseessä enemmänkin vapaaehtoinen auttaminen omasta puolestani. Minulle siis maksetaan projektissa kuvittajana.

Oma osani tarinan teossa on näkynyt Iltan luvussa sekä yleisesti pienessä avustamisessa tarinan kanssa. Omasta puolestani Iltta-luvun tekstin kanssa pätee samat säännöt kuin kuvienkin kanssa. Eli tarinaa käytetään projektissa, ja creditit tarinassa omasta puolestani ovat vain Iltan osiossa, opiskelijat ovat itse kirjoittaneet loput ja täten päättävät niihin koskevat säännöt yhdessä. Jos tarvetta, minulle voi antaa lisä-creditin tarinassa avustuksessa, mutta tämä ei ole välttämätöntä.

Opiskelijat ovat luoneet tarinoita luomieni/piirtämieni hahmojen ympärille. Opiskelijat ovat siis tekstin omistajia, enkä esimerkiksi somessa kuvia jakaessani yhdistä tekstiä konkreettisiin kuviin, tai väitä tekstejä omikseni. Poikkeuksena toki Iltan-teksti.

Jos näistä on mitään kysyttävää niin älkää arastelko kysymään! Kyseessä oli siis epävirallinen listaus, eli jos jotain on päässyt unohtumaan, minulle saa laittaa viestiä. Suurin osa näistä on jo melko päivänselviä asioita, mutta on hyvä että ne ovat muistissa niitä tarvittaessa!

Kuten jo aiemmin mainitsinkin, jos teillä on ideoita jatkoprojekteihin, lisäkuviin tai hahmojen ja kuvien käyttöön johonkin muuhun, siitä on mahdollista keskustella erillisenä työnä.

- Linnea Vaikkola 25.04.2023

Liite 5: Testauskysymykset

Mitä ajattelet pelistä?

Onko jotain mitä muuttaisit pelissä?

Tuleeko terveydellinen sisältö läpi pelaajalle?

Mitä tekisit toisin?

Mistä pidit pelissä?

Mitä mieltä olit pelin pituudesta?

Ilmenikö pelin aikana teknisiä ongelmia tai Bugeja?

Miten kuvailisit pelikokemusta?

Onko sinulla kommentoitavaa pelistä?

Vapaa sana....(risuja ja ruusuja)