MIIKKA REMES

Verkkopeliohjelmointi GameMaker: Studiolla

SÄHKÖ- JA AUTOMAATIOTEKNIIKAN
KOULUTUSOHJELMA
2022

Tekijä
Remes, Miikka

Julkaisun laji
Opinnäytetyö, AMK

Päivämäärä
Joulukuu 2022

Sivumäärä
31

Julkaisun kieli
Suomi

Julkaisun nimi
Verkkopeliohjelmointi GameMaker: Studiolla

Tutkinto-ohjelma
Sähkö- ja automaatiotekniikka

Tiivistelmä

Verkkopeliohjelmointi on aste eteenpäin pelien kehityksessä. Nykyään pelien kehitys onnistuu yksittäisiltä henkilöiltä sekä pieniltä tiimeiltä suhteellisen vaivattomasti nykyaikaisilla pelimoottoreilla. Verkkopeliohjelmointi tuo mukanaan uusia haasteita pelikehitykseen ja sen tuomat ominaisuudet voivat olla kovin tuottoisia.

Työn aiheena oli tutustua verkkopeliohjelmointiin luomalla yksinkertainen verkkopeli käyttäen GameMaker: Studio -pelimoottoria.

Työssä käsitellään verkkopeliohjelmoinnin eri osuuksia, sen toteutus tapoja ja keinoja optimoida verkkoliikennettä.

Asiasanat
Peliohjelmointi, pelimoottori, pelisuunnittelu

Author
Remes, Miikka

Type of Publication
Bachelor's thesis

Date
December 2022

Number of pages
31

Language of publication:
Finnish

Title of publication
Online game programming with GameMaker: Studio

Degree program
Electric- and Automation Engineering

Abstract

Online game programming is a step forward in game development. These days game development can be done relatively easily by individuals and small teams with modern game engines. Online game programming brings with it new challenges in game development, which can be very rewarding after getting through the new challenges brought by it.

The topic of the work was to get to know online game programming by creating a simple online game using GameMaker: Studio game engine.

In the work, the different parts of online game programming, its implementation methods and ways to optimize network traffic are discussed.

Key words
Game programming, Game engine, Game design

SISÄLLYS

1 JOHDANTO....5
2 VERKKOPELI-OHJELMOINTI....6
2.1 Pistoke ohjelmointirajapinta....6
2.2 Protokollat....6
2.2.1 TCP/IP....7
2.2.2 UDP/IP....7
2.3 Palvelintyypit verkkopeliohjelmoinnissa....8
3 GAMEMAKER: STUDIO....10
3.1 GameMaker Language Code....10
3.2 GameMaker Language Visual....11
3.3 Spritet....11
3.4 Ääniefektit....12
3.5 Huoneet....13
3.6 Oliot ja instanssit....14
3.7 Oliotapahtumat....15
3.7.1 Luomistapahtuma....15
3.7.2 Askeltapahtuma....15
3.7.3 Ajastintapahtuma....16
3.7.4 Tuhoamistapahtuma....16
3.7.5 Törmäystapahtuma....16
3.8 Verkkotapahtuma....16
4 PELISUUNNITELMA....19
4.1 Pelaajat....19
4.2 Tavoite....19
4.3 Tehosteet....19
4.4 Pelin kulku....20
5 PELIPROJEKTIN TOTEUTUS....21
5.1 Peliareena....22
5.2 Pelihahmojen ohjelmointi....23
5.3 Tehosteiden ohjelmointi....25
5.4 Pelin verkko-ohjelmointi....26
6 YHTEENVETO....28

LÄHTEET

1 JOHDANTO

Tässä opinnäytetyössä luotiin verkkopeli GameMaker: Studio pelimoottorilla, käyttäen sen sisältämää GameMaker Languagea ja sen ominaisuuksia. Työssä käytiin läpi verkkopeliohjelmoinnin periaatteita.

Verkkopelien suosio lähti nousuun 1990-luvun alkupuolella, kun internetin saatavuus kasvoi nopeasti. Tämän lisäksi internet nopeudet kasvoivat 1990-luvulla ja 2000-luvun alkupuolella, mikä nopeutti verkkopelien suosion kasvua. Ensimmäiset verkkopelit olivat suhteellisen yksinkertaisia tekstiseikkailupelejä. Verkkoyhteyksien nopeutuessa pelit ovat kehittyneet erilaisiin reaaliaikaisiin toiminta- ja seikkailupeleihin, joissa on erilaisia keskustelu mahdollisuuksia tekstin tai puheen kautta. Nykypäivänä on tuhansia erilaisia verkkopelejä, joista osa on valtavia ja kykenevät tukemaan palvelimillaan tuhansia pelaajia samanaikaisesti. (Datapath.io, 2017)

Opinnäytetyön peliä ohjelmoidessa konkretisoitui verkkopeliohjelmoinnin tuomat uudet haasteet. Opinnäytetyön aihe oli valittu allekirjoittaneen historiasta sekä kiinnostuksesta pelien kehitykseen, jolle verkkopeliohjelmointi oli uusi haaste ja askel eteenpäin peliohjelmoinnin parissa.

2 VERKKOPELI-OHJELMOINTI

GameMaker: Studio käyttää pistoke ohjelmointirajapintaa, joka mahdollistaa kahden sovelluksen keskinäisen keskustelun eri isäntäkoneilla.
Pistoke ohjelmointirajapinta on alan standardi kaikilla alustoilla. (YoYo Games Ltd, 2022j.)

2.1 Pistoke ohjelmointirajapinta

Pistokkeilla pystyy lähettämään, vastaanottamaan, yhdistämään ja kuuntelemaan portteja, mikä mahdollistaa kommunikaation palvelimen ja asiakkaan välillä helposti. Pistokkeet käsittelevät TCP/IP(Transmission Control Protocol / Internet Protocol) yhteyksiä, johon nykyään internet perustuu. Yksinkertaisesti pistokkeet sallivat yhteyden muodostamisen kahden pistokkeen välillä käyttäen IP-osoitetta. Yhteyden muodostamisen jälkeen, isäntäkoneet pystyvät lähettämään verkkopaketteja toisilleen pistokkeiden kautta. (The Open Group, 2018; YoYo Games Ltd, 2022j.)

2.2 Protokollat

Verkkopelien ohjelmoinnissa käytetään tyypillisesti TCP-protokollaa (*Transmission Control Protocol*) tai UDP-protokollaa (*User Datagram Protokollaa*), mutta suurin osa verkkopelin kommunikaatiosta tapahtuu käyttäen TCP-protokollaa. Verkkopeleissä missä verkkopaketteja täytyy lähettää jatkuvasti ja erittäin nopeasti, kannattaa käyttää UDP-protokollaa, jotta pakettien hallinta on kevyempää ja nopeampaa. TCP-protokollaa kannattaa käyttää esimerkiksi keskusteluikkunoissa ja muissa toiminnoissa, missä ei ole varaa menettää tietoja. (Caroll, 2022)

2.2.1 TCP/IP

Tyypillisesti kaikki tieto minkä on tärkeää saapua vastaanottavalle osapuolelle, lähetetään käyttäen TCP-protokollaa, sillä se sisältää mekanismeja vikatilanteista toipumiseen, varmistaa että paketit saapuvat perille sekä varmistaa, että paketit saapuvat oikeassa järjestyksessä. Tarpeen tullessa paketti lähetetään uudelleen, jos se ei ole saapunut perille. TCP-yhteys sisältää kolme eri vaihetta, yhteyden muodostaminen, tiedonsiirto sekä yhteyden katkaisu. (Postel, 1983, s. 2)

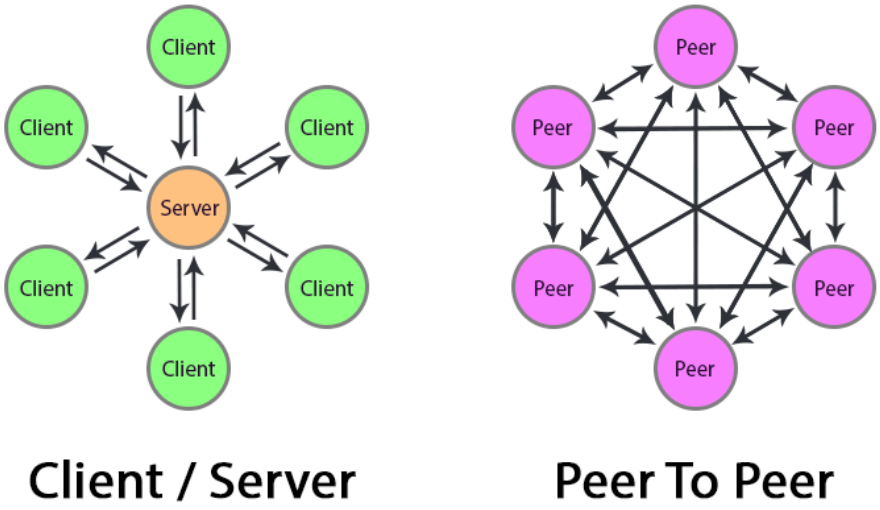
TCP-yhteyden muodostamiseen käytetään kolmitiekättelyä. Kättelyssä yhteyden asiakas lähettää SYN-paketin(Synchronization) palvelimelle, joka vastaa lähettämällä SYN-ACK-paketin(synchronization-acknowledgement), johon asiakas viimeiseksi vastaa ACK-paketilla(acknowledgement). Tiedonsiirrossa on käytössä monia mekanismeja, jotka varmistavat verkkopaketin eheyden ja saapumisen perille. TCP-pakettien järjestyksen varmistamiseksi käytetään sekvenssinumerointia. Virheiden tarkistusta varten käytetään tarkistussummia. Hukatuilla paketeille ja viiveelle käytetään ajastimia ja tunnistimia. Tietovirrat tunnistetaan toisistaan yhteyden muodostuksen yhteydessä lähetetystä sekvenssinumerosta, joiden avulla tunnistetaan kullekin yhteydelle kuuluvat paketit. Mikäli kuittausta lähetetylle paketille ei saada, paketti lähetetään uudelleen. Yhteyden katkaiseminen tapahtuu joko nelitiekäsittelyllä tai kolmitiekäsittelyllä. Nelitiekäsittelyssä osapuolet katkaisevat yhteyden erikseen lähettämällä FIN-paketin ja kuittaamalla sen ACK-paketilla. Kolmitiekäsittelyssä toinen osapuoli lähettää FIN-paketin, johon toinen osapuoli vastaa FIN-ACK paketilla. Tämän jälkeen ensimmäinen osapuoli vastaa FIN-ACK pakettiin ACK-paketilla. Näiden lisäksi on myös olemassa keino katkaista yhteys välittömästi, lähettämällä RESET-valitsimella varustettu paketti, johon ei lähetetä vastaus paketteja. (Verma, 2022.)

2.2.2 UDP/IP

User Datagram Protokolla on yksinkertainen datagram-pohjainen kuljetuskerroksen protokolla, jossa ei muodosteta yhteyttä, vaan verkkopaketti lähetetään vastaanottajalle varmistamatta, onko se vastaanotettu onnistuneesti. UDP-protokolla ei sisällä minkäänlaisia tarkastuksia verkkopakettien ruuhkanhallintaan, kaksoiskappaleiden poistoon tai vuonhallintaan. UDP-protokolla eroaa TCP-protokollasta monella tapaa. UDP-protokollan yleisrasite on pienempi, koska siinä ei suoriteta alkukäsittelyä, pakettien kuittausta tai yhteyden lopettamista, mikä tekee siitä joissain tapauksissa nopeamman kuin TCP-protokolla. Verkkopeliohjelmoinnissa UDP-protokollaa voi käyttää erilaisten tietojen lähettämiseen, joiden perille saapumisen menetyksellä silloin tällöin ei ole suurta merkitystä pelin toiminnan kannalta. (Imperva Inc, 2021)

2.3 Palvelintyypit verkkopeliohjelmoinnissa

Verkkopeliohjelmoinnissa on pääasiassa kaksi eri tapaa luoda palvelimia. Oma palvelin on palvelintyyppi, missä palvelin on erillään pelaajista ja hallitsee vain verkkopakettien vastaanottamisen ja lähettämisen asiakkaiden välillä (Kuva 1). Tällaisessa järjestelyssä kaikki pelaajat yhdistävät palvelimeen, eivätkä ole suorassa yhteydessä toisiinsa. Oma palvelin on paras vaihtoehto, kun peli on suurikokoinen ja saattaa aiheuttaa performanssi ongelmia tavallisilla tietokoneilla. Toinen vaihtoehto on niin kutsuttu all-in-one-palvelin, jota usein kutsutaan ”peer to peer” -menetelmäksi. Tässä menetelmässä peliä isännöi yksi pelaaja, mutta kaikki peliin liittyvät pelaajat ovat yhteydessä suoraan toisiinsa (Kuva 1). (YoYo Games Ltd., 2019.)



Kuva 1. Palvelintyypit (YoYo Games Ltd., 2019.)

3 GAMEMAKER: STUDIO

GameMaker: Studio on alun perin 1999-luvulla Mark Overmarsin kehittämä monitasoinen pelimoottori, jonka kehitystä on jatkanut YoYo Games vuodesta 2007 eteenpäin. (Center for Digital Games Research, 2022) GameMaker: Studio houkuttelee pelikehittäjiä useammalta ammattitaitotasolta, sillä GameMakeristä löytyy visuaalinen raahaa ja pudota ohjelmointikieli aloitteleville pelikehittäjille, sekä GameMaker Language -komentosarjakieli kehittyneemmille pelikehittäjille. (YoYo Games Ltd, 2022k.)

3.1 GameMaker Language Code

GameMaker Language on imperatiivinen ja dynaamisesti tyypittävä komentosarjakieli. GML koodia voidaan käyttää yhdessä visuaalisen GML kanssa. Koodin täytyy sisältää perusrakenne ja se voi sisältää resurssi-indeksejä, muuttujia, funktioita, ilmaisuja ja avainsanoja. Koodi kirjoitetaan lauseissa, jotka voivat olla yksinkertaisia kuten kahden luvun yhteenlasku tai monimutkaisia kuten pelihahmon käyttäytyminen erilaisissa tilanteissa. Koodilauseet tulisi erotella toisistaan ";"-symbolilla, jotta koodi pysyisi siistinä ja virheiltä vältyttäisiin. Lauseet voivat koostua muuttujien luonnista, lausekkeista ja funktioiden kutsuista. Koodin rakenne voi vaihdella suuresti riippuen käytetyistä funktioista, mutta koodi rikotaan aina perusteisiin (Kuva 2). Lauseita voi myös ryhmittää käyttäen aaltosulkeita, jolloin niitä voidaan kutsua yhdessä (Kuva 3). (YoYo Games Ltd, 2022a.)

```
<statement>;
<statement>;
...
```

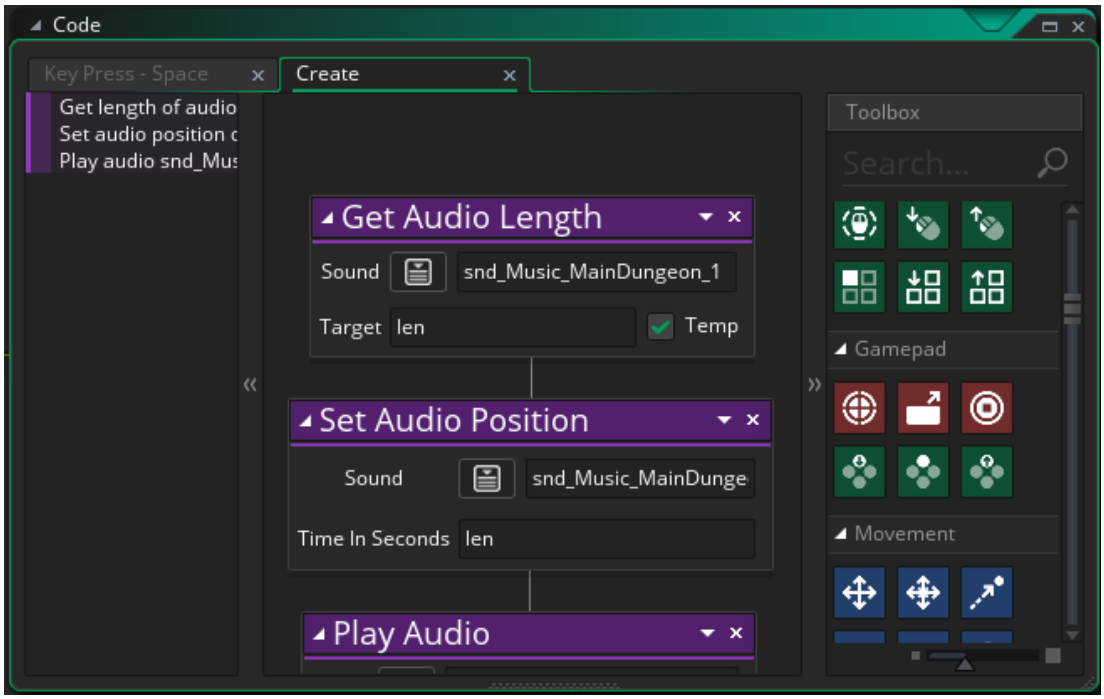
Kuva 2. Koodin rakenne (YoYo Games Ltd, 2022a.)

```
if (<conditional>)
{
    <statement>;
    <statement>;
    ...
}
```

Kuva 3. Ryhmitettyjä lauseita (YoYo Games Ltd, 2022a.)

3.2 GameMaker Language Visual

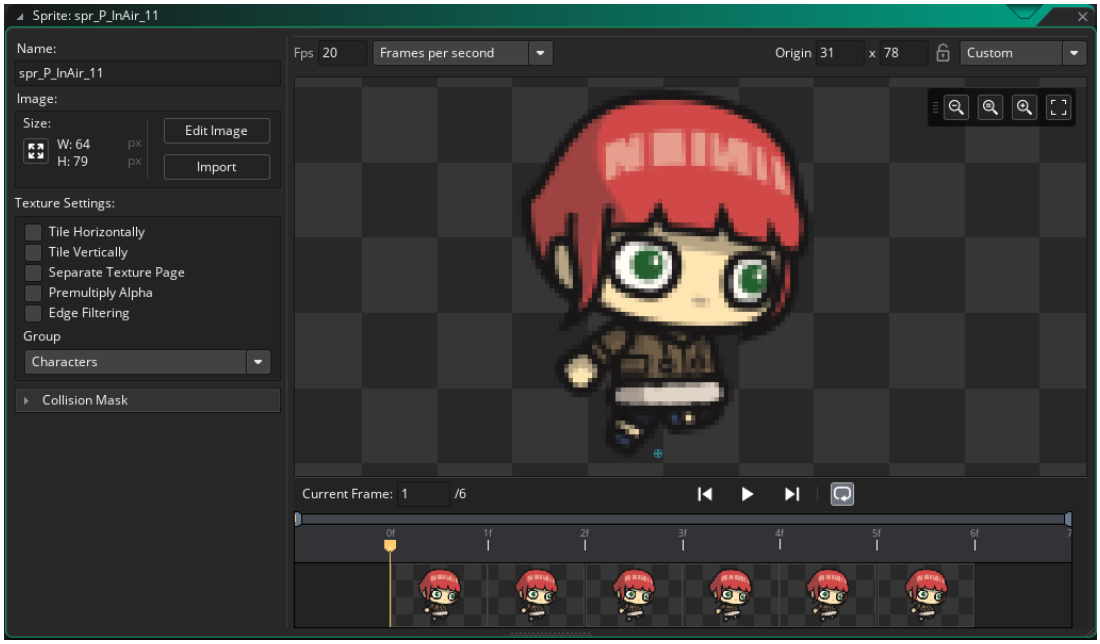
GameMaker Language Visual on komentosarja työkalu, jonka avulla pelejä voi tehdä kirjoittamatta koodia itse. Visuaalisella GameMaker Languagella koodi luodaan käyttäen visuaalisia toiminta palikoita, joita voidaan käyttää ketjussa luoden useita toimintoja eri tilanteissa. Esimerkiksi olion luonnin yhteydessä voidaan yhdellä toiminto palikalla saada ääniefektin pituus, toisella asettaa äänen sijainti ja kolmannella soittaa kyseinen ääni (Kuva 4). (YoYo Games Ltd, 2022b.)



Kuva 4. Koodi ikkuna (YoYo Games Ltd, 2022b.)

3.3 Spritet

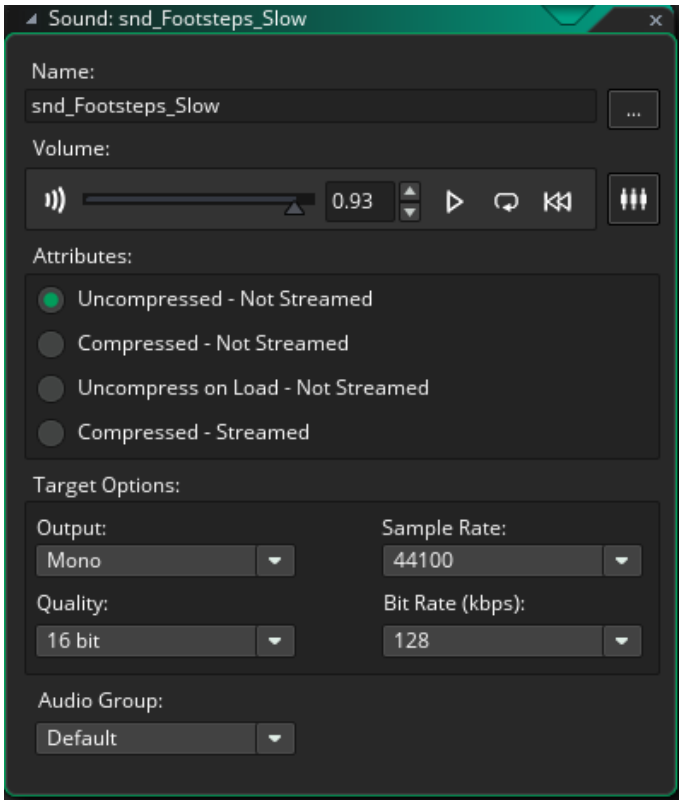
GameMakerissä käytetään spritejä olioiden kuviksi. GameMaker sisältää kuvan editointityökalun, jolla voi luoda, muokata ja animoida spritejä, sekä määrittää niiden pinnan, mitä pelissä käsitellään kuvan pintana törmäyksen tunnistukseen (Kuva 5). (YoYo Games Ltd, 2022c.)



Kuva 5. Spriten editointi ikkuna (YoYo Games Ltd, 2022c.)

3.4 Ääniefektit

Kaikissa peleissä tarvitaan ääniefektejä ja musiikkia. GameMakerissa ääniä pystytään lisäämään peliin käyttäen sen ääni editoria, jolla voi avata äänitiedostoja (Kuva 6). Äänitiedostot pystytään käsittelemään kompressoituna tai kompressoimatta, joista kompressoituja ääniä pystytään käyttämään monipuolisemmin. Kompressoimattomia äänitiedostot ovat tallennettu WAV-formaattiin, kun taas kompressoidut äänitiedostot ovat MP3- tai OGG-formaatissa. WAV-formaattiin tallennetut tiedostot kannattaa pitää pieninä ääniefekteinä. Suuremmat äänitiedostot kannattaa tallentaa MP3- tai OGG-formaattiin. GameMaker Language sisältää myös funktioita, joilla ääniin pystyy lisäämään efektejä pelin pyöriessä. (YoYo Games Ltd, 2022d.)



Kuva 6. Ääniefektin editointi ikkuna (YoYo Games Ltd, 2022d.)

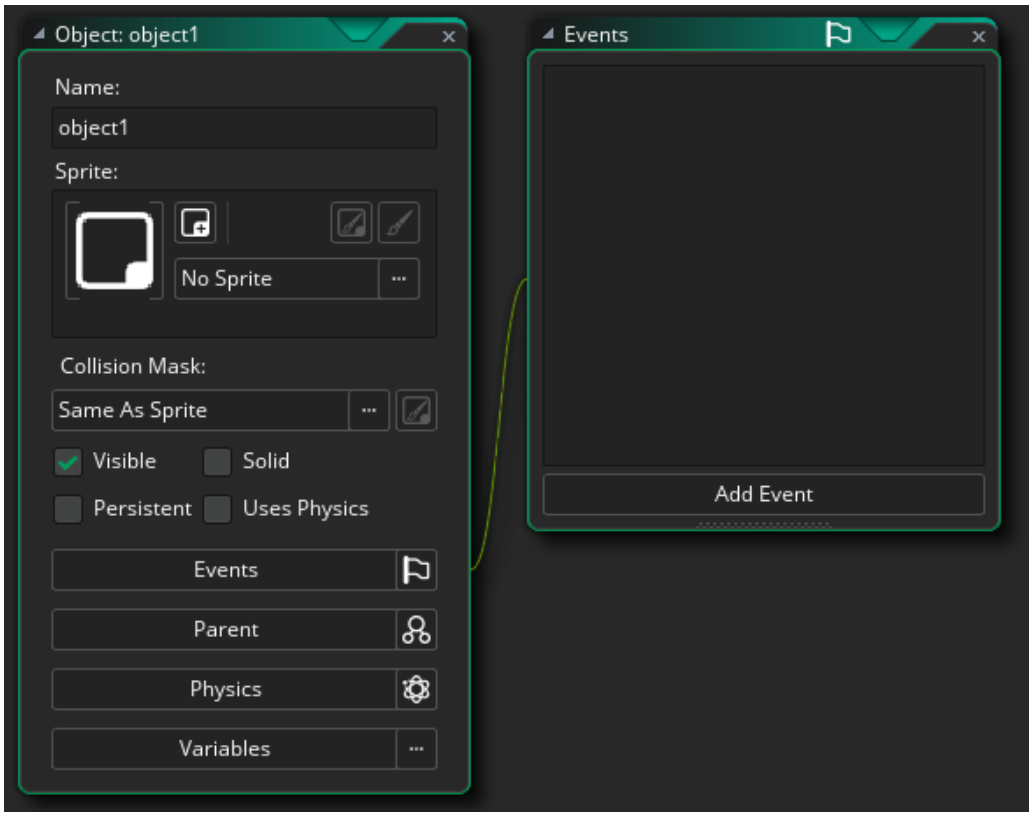
3.5 Huoneet

GameMakerissa pelit tarvitsevat aina vähintään yhden huoneen. Huone on yksinkertaisesti tila, minne pelin instansseja asetetaan huone editorilla (Kuva 7). Huoneeseen lisätyt instanssit menevät kerroksiin, jotka määrittävät mikä niistä piirretään päällimmäiseksi. Huoneeseen määritellään myös kamera-asetukset, mitkä määrittävät minkä kokoinen alue huoneesta piirretään ja kuinka suurelle ikkunalle.

Kuva 7. Huone-editori ikkuna (YoYo Games Ltd, 2022e.)

3.6 Oliot ja instanssit

GameMakerissa käytetään olioita ja instansseja. Oliot ovat käytännössä pohjia instansseille, eivätkä ne ole olemassa pelin huoneessa, ennen kuin niistä tehdään instanssi. GameMakerissa koodia ja toiminnallisuuksia lisätään olioeditorissa olioiden eri tapahtumille (Kuva 8). Olioilla on eri tapahtumia, joita käydään läpi pelin pyöriessä järjestyksessä. Esimerkiksi ensimmäisenä oliossa käydään läpi luomistapahtuma, jonka sisälle voi lisätä erilaisia funktioita ja koodia, joilla voidaan alustaa esimerkiksi elämät, pisteet, nopeus ja sijainti. Askeltapahtuma käydään läpi jokaisella pelin kierroksella, joten tähän tapahtumaan lisätään suurin osa pelin toiminnallisuuksista, kuten esimerkiksi olion liikuttaminen ja näppäin painallusten toiminnot. (YoYo Games Ltd, 2022f.)



Kuva 8. Olioeditori ikkuna (YoYo Games Ltd, 2022f.)

3.7 Oliotapahtumat

Oliotapahtumat ovat erilaisia hetkiä pelissä, joita suoritetaan eri kohdissa peliä ja ne suorittavat sen mitä niihin on ohjelmoitu. (YoYo Games Ltd, 2022g.)

3.7.1 Luomistapahtuma

Olion luomistapahtuma ajetaan vain yhden kerran. Luomistapahtuma ajetaan silloin kun olio luodaan huoneeseen, mikä tekee siitä hyvän paikan missä alustaa muuttujia, ajastimia tai mitä tahansa mikä tarvitsee tehdä vain yhden kerran. (YoYo Games Ltd, 2022g.)

3.7.2 Askeltapahtuma

Olioilla on askeltapahtuma, mikä suoritetaan jokaisella ohjelmakierroksella. Pelin ohjelmakierroksen nopeuden pystyy asettamaan tiettyyn arvoon

peliprojektin asetuksista. Esimerkiksi pelin nopeus asetettuna arvoon 60, peli pyörii 60-kierrosta sekunnissa, joista jokaisella kierroksella käydään läpi kaikkien olioiden askeltapahtuma. (YoYo Games Ltd, 2022g.)

3.7.3 Ajastintapahtuma

Ajastintapahtumat ovat kätevä keino suorittaa asioita tietyn ajan. Ajastimet seuraavat ohjelmakierrosta, esimerkiksi jos pelin nopeus on 60 ja ajastimeen asettaa 60, tapahtuma suoritetaan yhden sekunnin päästä.
Ajastintapahtumassa ajastin voidaan myös asettaa uudelleen, jolloin ajastimesta saadaan jatkuvasti tietyn aikavälein suoritettava tapahtuma, mikä on kätevää esimerkiksi verkkopeliohjelmoinnissa silloin, kun tietoa ei tarvitse lähettää joka ohjelmakierroksella. Tällein voi säästää lähetettävän tiedon määrässä ja vähentää sen aiheuttamaa kuormaa. (YoYo Games Ltd, 2022g.)

3.7.4 Tuhoamistapahtuma

Tuhoamistapahtuma suoritetaan silloin kun olion instanssia ollaan tuhoamassa. Tuhoamistapahtuman avulla voidaan esimerkiksi luoda erilaisia efektejä tai tapahtumia olion instanssin tuhoutuessa, kuten pisteiden lisääminen ja uuden olion instanssin luominen. (YoYo Games Ltd, 2022g.)

3.7.5 Törmäystapahtuma

Peleissä on tärkeää tietää, milloin jokin olion instanssi on törmännyt toiseen instanssiin. Törmäystapahtuma tarkistaa törmäystä määriteltyihin olioihin.
Törmäystapahtuma ajetaan jokaisella ohjelma syklillä. Tapahtumaa voidaan käyttää esimerkiksi havaitsemaan seiniin törmäys ja pelaajan hahmoa vahingoittaviin asioihin törmäys. (YoYo Games Ltd, 2022g.)

3.8 Verkkotapahtuma

GameMakerin verkkotapahtuma laukeaa kaikesta vastaanotettavasta verkko liikenteestä. GameMaker sisältää funktioita, joiden avulla palvelimen luonti ja siihen yhdistäminen asiakkaana on yksinkertaista. Verkkofunktiot perustuvat pistokkeisiin ja palvelimen luominen onnistuu yhdellä funktion kutsulla, joka palauttaa pistokkeen, minkä avulla kommunikointi onnistuu helposti asiakkaiden kanssa. Funktio ottaa argumenteiksi yhteys tyypin, portin ja maksimi asiakkaiden määrän (Ohjelma 1). (YoYo Games Ltd, 2022h.)

```
socket = network_create_server(type, port, max_client);
```

Ohjelma 1. Palvelimen luomisfunktio

Pistokkeet käyttävät TCP/IP yhteyttä, joten yhdistäminen palvelimeen onnistuu käyttäen IP-osoitetta. Luotuun palvelimeen pystyy yhdistämään luomalla pistokkeen funktiolla, joka ottaa argumentiksi vain yhteys tyypin ja käyttämällä yhdistysfunktiota, joka ottaa argumenteiksi pistokkeen, IP-osoitteen ja portin (Ohjelma 2). (YoYo Games Ltd, 2022h.)

```
socket = network_create_socket(network_socket_tcp);
network_connect(socket, url, port);
```

Ohjelma 2. Pistokkeen luominen ja yhdistys palvelimeen

Muodostettuaan yhteyden, argumentiksi annettua pistoketta voi käyttää kommunikointiin palvelimen kanssa. Verkkopaketin lähettäminen pistokkeiden välillä onnistuu funktiolla mikä ottaa argumenteiksi vain pistokkeen tunnuksen, lähetettävän verkkopaketin tunnuksen ja verkkopaketin koon (Ohjelma 3). (YoYo Games Ltd, 2022h.)

```
network_send_packet(socket, bufferid, size);
```

Ohjelma 3. Verkkopaketin lähetysfunktio

Vastaanotettu verkkopaketti laukaisee verkkotapahtuman, minkä avulla vastaanotettu verkkopaketti tiedetään käsitellä. Verkkopaketin tietotyyppi on puskuri, joka on vain varattu muistialue tiedolle. Puskureita voi luoda erilaisia,

kuten tietyn kokoisia tai dynaamisesti kokoaan kasvattavia (Ohjelma 4). (YoYo Games Ltd, 2022h.)

```
buffer_create(size, type, alignment);
```

Ohjelma 4. Puskurin luomisfunktio

Parhaimman performanssin saa puskurilla, jonka koko ei muutu dynaamisesti, varsinkin siinä tapauksessa, kun puskuriin kirjoitetaan tietoa jatkuvasti nopealla tahdilla. Puskurille myös annetaan argumentiksi tavu kohdistus, mikä muuttaa kirjoitettavan tiedon kohdistuksen (Kuva 9). Esimerkiksi yhden tavun kohdistus tarkoittaa, että kaikki tieto puskuriin kirjoitetaan heti edellisen tiedon perään. Kahden tavun tahdistus tarkoittaa, että tieto kirjoitetaan kahden tavun intervalleissa, eli vaikka kirjottaisit vain yhden tavun verran tietoa, seuraava tiedon kirjoitus siirretään vastaamaan kahden tavun tasausta. Yleensä yhden tavun kohdistus toimii, mutta esimerkiksi jos puskuriin kirjoitettava tieto on aina 16-bittisiä lukuja, tulisi kohdistuksen olla kaksi tavua. (YoYo Games Ltd, 2022i.)

Kuva 9. Puskurin kohdistus (YoYo Games Ltd, 2022i.)

4 PELISUUNNITELMA

Peliksi on suunniteltu kaksiulotteinen ottelupeli, jossa pelaajat pelaavat käärmemäisillä olennoilla. Pelin tarkoituksena on selviytyä pisimpään törmäämättä käärmeiden jatkuvasti kasvaviin häntiin, hyödyntäen ympäristöön ilmestyviä tehosteita. Törmätessään häntään pelaajan käärme kuolee, josta kaikki elossa olevat käärmeet saavat pisteen, näin ollen pisimpään selviytynyt kerää eniten pisteitä.

4.1 Pelaajat

Pelissä pelaajat ohjaavat käärmeitä, jotka liikkuessaan tuottavat jatkuvasti häntää, jättäen siihen satunnaisesti aukkoja, joista käärmeellä mahtuu liikkumaan lävitse. Käärmeet ovat tavallisesti jatkuvasti liikkeessä, jota pelaaja voi ohjata vain kääntämällä käärmeen suuntaa oikealle tai vasemmalle, esimerkiksi nopeutta pelaaja ei voi säätää.

4.2 Tavoite

Pelin tavoitteena on kerätä pisteitä selviytymällä pisimpään. Pelin voittaa pelaaja, joka kerää ensimmäisenä määrätyn määrän pisteitä, joka vaihtuu sen mukaan, kuinka monta pelaajaa pelissä on, vaihtuen kahden pelaajan 10-pisteen tavoitteesta kuuden pelaajan 30-pisteen tavoitteeseen.
Pisteiden tavoittelemiseksi voi esimerkiksi pyrkiä saartamaan muita pelaajia hännästä muodostuneisiin nurkkiin tai keskittyä vain selviytymiseen ja etsiä reittiä läpi esteiden mahdollisimman pitkälle.

4.3 Tehosteet

Pelissä ilmestyy ympäristöön satunnaisesti rajatun aikavälein erilaisia tehosteita, joita keräämällä voidaan saada käärmeelle erilaisia hyödyllisiä tai haitallisia avustuksia, kuten väliaikainen nopeuden kasvatus tai -hidastus,

muiden pelaajien nopeuden kasvatus tai -hidastus, hetkellisesti kokonaan pysähtyminen tai tehoste, jolla väliaikaisesti pääsee käärmeiden häntien lävitse.

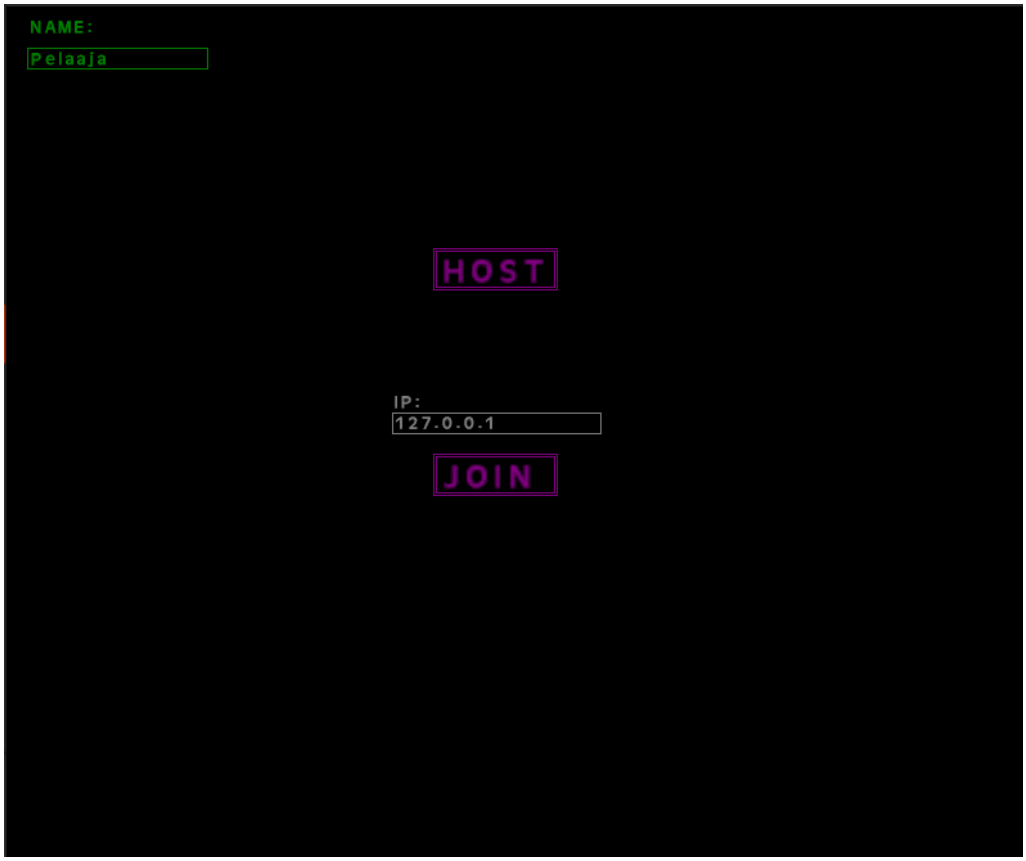
4.4 Pelin kulku

Avatessa pelin, avautuu päävalikko, missä on pelin titteli, tekstikenttä pelaajan nimelle sekä IP-osoitteelle, mitä käytetään, mikäli pelaaja valitsee yhdistämisen palvelimeen asiakkaana. Valikosta peli käynnistetään valitsemalla, yhdistetäänkö palvelimeen vai luodaanko uusi peli. Päävalikossa annetaan virheilmoitus, jos syötetty IP-osoite ei ole kelvollinen tai jos yhteyttä ei pystytty muodostamaan.

Pelin käynnistäminen palvelimena aukaisee peliareena huoneen, missä itse toiminta tapahtuu. Huone on alussa pysähdyksissä, kunnes pelaaja painaa välilyöntiä. Huoneen ollessa pysähdyksissä alussa, peli sallii uusien pelaajien yhdistää palvelimeen, jotka yhdistäessään saavat pelihahmoinstanssin ohjattavaksi huoneeseen. Asiakkaana peliin yhdistäessä siirrytään myös peliareena huoneeseen, jossa jäädään vastaavasti odottamaan palvelimelta tietoa, että pelin luoja on käynnistänyt pelin. Pelaajat saavat palvelimelta tiedon muiden pelissä olevien hahmojen sijainnista ja suunnasta. Jokaisen pelaajan huoneeseen lisätään instanssit muista pelaajista. Pelissä voittajan kerättyä voittamiseen vaadittavat pisteet saa pelin pysähtymään ja pelaajille tulee ilmoitus voittajasta. Tämän jälkeen pelistä voi poistua painamalla välilyöntiä. Mikäli palvelin lopettaa pelin kesken kaiken, muut pelaajat siirtyvät päävalikkoon.

5 PELIPROJEKTIN TOTEUTUS

Suunniteltu peliprojekti vaatii toimiakseen huoneet päävalikolle ja peliareenalle, olion pelaajien ohjattavaksi, spritejä hahmoille, tehosteille ja efekteille, ääniefektit ja ohjaus olio, joka hoitaa verkkoyhteydet ja kommunikaation pelien välillä. Luon pelille päävalikoksi yksinkertaisen huoneen huone-editorilla mustalla taustalla, johon asetan näkymättömän ohjain olion. Ohjain olio piirtää ja hallitsee huoneeseen tekstikentät pelaajan nimelle ja IP-osoitteelle. Tämän lisäksi olio piirtää kaksi näppäintä, joita painamalla voi joko aloittaa pelin palvelimena tai liittyä toisen pelaajan luomaan peliin (Kuva 10). IP-osoitteen tekstikenttään ei sallita muuta kuin ip-osoitteeseen kuuluvia merkkejä.



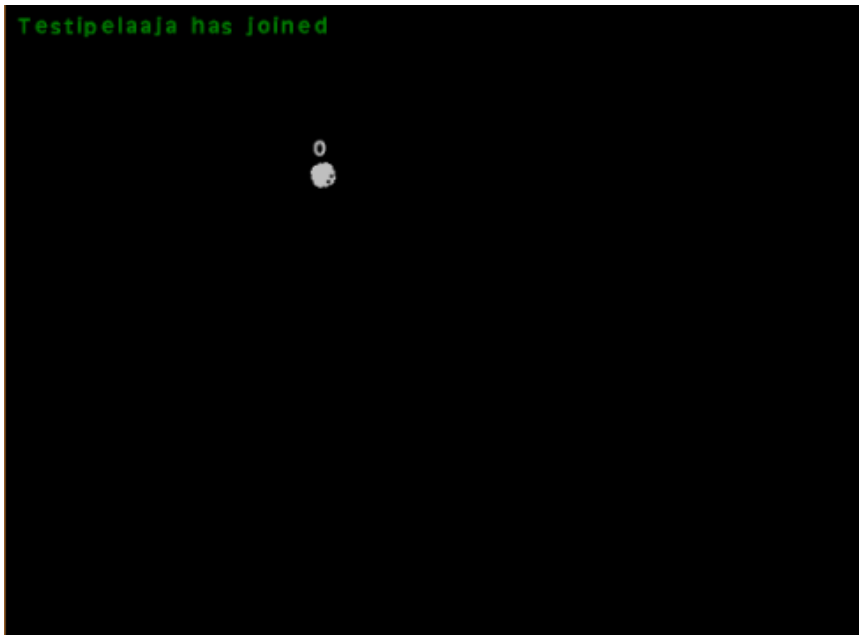
Kuva 10. Päävalikko

5.1 Peliareena

Peliareenana toimii huone, missä toimii taustana pelkkä musta tausta, mikä pitää huoneen selkeän näköisenä kaikkien huoneessa olevien instanssien kanssa, sillä peliareena täyttyy nopeasti käärmeiden hännistä, verestä ja tehosteista (Kuva 11). Huoneen ominaisuuksia on huoneen koko, mikä tässä pelissä voisi olla muutettavissa, tuoden lisä haasteita pelaajille. Huoneeseen ei sijoiteta muuta kuin pelin ohjausinstanssi, mikä hallitsee pelihahmoinstanssien luonnin huoneeseen. Huoneen reunojen läpi pystyy kulkemaan, jolloin pelaajan hahmo ilmestyy vastakkaiselle puolelle huonetta.

Kuva 11. Peliareena

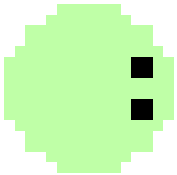
Muiden pelaajien liittyessä huoneeseen, tulee siitä pelissä oleville pelaajille ilmoitus peli-ikkunan yläkulmaan (Kuva 12). Kun pelaaja on liittynyt, hänelle luodaan hahmoinstanssi ohjattavaksi jokaisessa pelissä.



Kuva 12. Ilmoitus pelaajan liittymisestä

5.2 Pelihahmojen ohjelmointi

Pelaajan hahmona toimii käärme, jonka spriteksi tulee pelkkä pää. Jotta käärmeen suunnan tunnistaisi, piirsin päälle silmät (Kuva 13). Kuva on vain 16 pikseliä leveä ja 16 pikseliä korkea ja se on valkoinen, josta värin pystyy muuttamaan eri pelaajille koodista.



Kuva 13. Käärmeen sprite

GameMakerissä on valmis funktio mistä saa olion suunnan, minkä avulla ohjelmoin spriten kääntymään liikkumissuuntaan, käyttäen spriten piirtämisfunktiota ja käyttämällä suuntaa spriten kierron argumenttina (Ohjelma 5).

```
draw_sprite_ext(spr_kaarme, 0, x, y, 1, 1, suunta, vari, 1);
```

Ohjelma 5. Spriten piirtämisfunktioon voidaan antaa useita argumentteja

Käärmeen ohjaaminen onnistuu nuoli näppäimillä, joiden painallusta tarkastellaan jokaisella olion askeltapahtumalla. Näppäimen painallus kasvattaa tai pienentää olion suuntaa, mikä saa sen kääntymään haluttuun suuntaan (Ohjelma 6).

```
if keyboard_check(vk_left)
{
   direction += turn_speed;
}
if keyboard_check(vk_right)
{
   direction -= turn_speed;
}
```

Ohjelma 6. Spriten piirtämisfunktioon voidaan antaa useita argumentteja

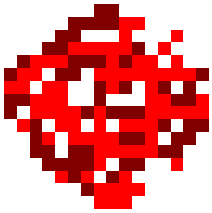
Käärmeet kasvattavat häntäänsä jatkuvasti. Hännälle spriteksi tein samanlaisen ympyrän kuin päästä, mutta ilman silmiä (Kuva 14). Hännän väri säädetään myös koodista spriten piirtämisfunktion argumenttina.

Kuva 14. Hännän sprite

Pelissä käärmeet kasvattavat häntäänsä jatkuvasti, joka koostuu yllä mainituista ympyröistä. Tähän käytin ajastinfunktioita, jotka luovat häntäinstansseja jälkeensä liikkuessaan, pitäen muutaman häntä kappaleen taukoja satunnaisen ajan välein.

Kaikkiin käärmeen ajastimiin ja nopeuksiin on kerroin, mikä voi muuttua peliareenasta kerätyillä tehosteilla. Kerroin tulee lisätä kaikkiin olion ominaisuuksiin, jotta käärme yhä kääntyy ja liikkuu oikeaa vauhtia. Kertoimella myös varmistetaan, että häntä kasvaa oikeaa tahtia jättämättä liian isoja tai pieniä rakoja.

Käärmeen törmäämistä tarkkaillaan jokaisella pelin kierroksella törmäystapahtumassa. Törmäystä tarkkaillaan kaikkien pelaajien häntiin, törmätessä käärme kokee verisen kuoleman. Kuolema efektiksi tein muutaman pikselin kokoisen veri spriten, josta tein olion, mikä saa luomishetkellä satunnaiseen suuntaan satunnaisen määrän nopeutta. Kuollessa veri oliosta tehdään 200 instanssia kuolleen käärmeen kohdalle, joka saa kuoleman näyttämään erityisen graafiselta. Veri olion törmätessään häntään saa hännän spriten muuttumaan verisemmäksi (Kuva 15).



Kuva 15. Verinen häntä sprite

5.3 Tehosteiden ohjelmointi

Tehosteet ovat instansseja, jotka eivät itsessään sisällä tapahtumia. Tehosteita ilmestyy peliareenalle kymmenen sekunnin välein ja ilmestyvä tehoste on satunnainen. Tehosteilla on tyyppi, mistä niitä keräävä käärme tunnistaa mitä sen keräämisen seurauksena tapahtuu. Pelissä nopeutta muuttavat tehosteet vaikuttavat väliaikaisesti pelaajan käärmeen nopeus kertoimeen. Oman käärmeen nopeutta nostava tehosteen sprite on salama, kokonaan käärmeen pysäyttävään tehosteen kuva on tiimalasi ja nopeutta alentavan tehosteen sprite on etana (Kuva 16). Muiden pelaajien nopeutta muuttava tehoste aiheuttaa verkko paketin lähetyksen jokaiselle pelaajalle palvelimen kautta, jossa ilmoitetaan mikä tehoste pelaajille tulee voimaan. Muihin pelaajiin kohdistuvat tehosteet omaavat punaisen taustan, mutta kuva on muuten sama kuin omaan käärmeeseen kohdistuva tehoste (Kuva 16). Sinisellä taustalla olevat tehosteet kohdistuvat peliareenaan, eikä pelaajiin. Tehoste, minkä sprite on kolme mustaa pistettä, lisää tehosteiden ilmestymistä yhdellä tehosteella jokaiselle tehosteen ilmestymistapahtumalle (Kuva 16).

Kuva 16. Kokoelma tehoste spritejä

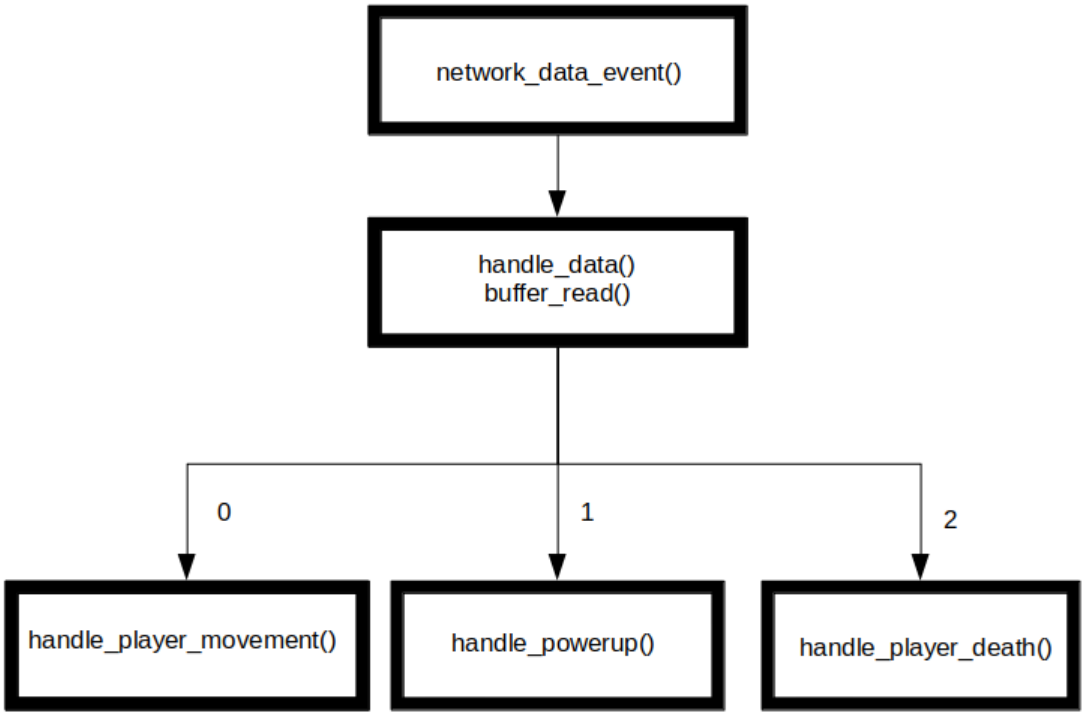
5.4 Pelin verkko-ohjelmointi

Pelin ohjain hallitsee monia asioita. Se hallitsee palvelimen tai asiakkaan luomisen, instanssien luomisen huoneeseen, taustamusiikin, pisteytyksen ja pelin päättymisen. Pelin ohjain luo valinnasta riippuen palvelin tai asiakas olion, joka kuuntelee verkkoliikennettä ja käyttää useita funktioita hallitakseen tapahtumia. Peliin päätin luoda oma palvelin pohjaisen toteutuksen, mutta en luonut erillistä projektia palvelinta varten, vaan palvelimena toimii sama peli sovellus, missä itse peli pelataan. Palvelin olio siis hallitsee kaikkien peliin liittyneiden asiakkaiden tiedon hallinnan, vastaanottaa asiakkaiden verkkopaketit, jälleen lähettää verkkopaketteja muille asiakkaille sekä ilmoittaa palvelin pelaajan omista tapahtumista muille.

Asiakkaiden hallinta onnistuu kätevästi tallentamalla jokaisen asiakkaan liittyessä pistoke tunnus ja käyttämällä sitä tunnistamaan mihin asiakkaaseen tapahtumat kohdistuvat. Aina uuden asiakkaan liityttyä pistoke tunnus

välitetään muille asiakkaille ja samaa tunnusta käytetään verkkopakettien puskurissa pelaajan tunnistukseen, jota tapahtuma koskee.

Verkkopakettien puskurin sisällön tunnistamiseksi ensimmäiseen tavuun kirjoitetaan tunnus luku, jolla vastaanotetun verkkopaketin käsittely voidaan jakaa funktioihin, missä käsitellään erilaisia tapahtumia tunnuksesta riippuen (Kuva 17). Esimerkiksi jos kyseessä on pelaajan käärmeen sijainti ja suunta tiedosta, ensimmäinen tunnus on 0, jonka avulla ohjataan funktioon, mikä purkaa loput puskurin tiedoista päivittääkseen oikean pelaajan käärmeen instanssin sijainnin ja suunnan. Tunnuksella 1 ohjataan funktioon, missä tiedetään puskurin seuraavan tavun olevan toinen tunnus luku, jonka avulla tiedetään mikä tehostetapahtuma käsitellään. Jos puskurin ensimmäisenä tunnuksena on 2, käsitellään jonkin pelaajan kuolema ja tästä jaetaan pisteet elossa oleville pelaajille. Tunnuksia on lukuisia ja samalla periaatteella saadaan käsiteltyä kaikki tarvittavat tapahtumat yksinkertaisesti.



Kuva 17. Verkkotapahtuman käsittelyn visualisointi

6 YHTEENVETO

Tein verkkopelin GameMaker: Studiolla hyödyntäen sen verkko-ominaisuuksia ja funktioita. Verkko osuuden ohjelmointi vaati hieman suunnittelua ja mietiskelyä, kuinka verkkopaketit saadaan jokaiselta pelaajalta jokaiselle muulle pelaajalle yksinkertaisesti ja skaalattavasti, jotta uusien toiminnallisuuksien lisääminen peliprojektiin olisi kätevää. Verkkopelejä ylipäätään ohjelmoidaan samalla periaatteella, joten tämän verkkopeliprojektin jälkeen on helppo ryhtyä suurempien verkkopeliprojektien pariin muillakin pelimoottoreilla. Syy miksi tein pelin pelimoottorilla GameMaker: Studio on, että olen käyttänyt pelimoottoria aiemmin yksinpelien luomiseen harjoitellessani pelikehitystä. GameMaker: Studio on joustava pelimoottori monenlaisia peliprojekteja varten ja saatan jatkossakin aloittaa uusia projekteja käyttäen sitä, jos kyseessä on kaksi-ulotteinen peli. Johtopäätöksenä GameMaker: Studio on varsin hyvä työkalu itsenäiselle pelikehittäjälle suhteellisen pieniin peliprojekteihin.

LÄHTEET

YoYo Games Ltd. (2019a), Beginners Guide To Networking, Haettu: 25.11.2022 osoitteesta https://gamemaker.io/en/blog/beginners-guide-to-networking

YoYo Games Ltd. (2022a), GameMaker Manual Basic Code Structure. Haettu 15.11.2022 osoitteesta https://manual.yoyogames.com/#t=GameMaker_Language%2FGML_Overview%2FBasic_Code_Structure.htm

YoYo Games Ltd. (2022b), GameMaker Manual GML Visual Overview. Haettu 18.11.2022 osoitteesta https://manual.yoyogames.com/index.htm?rhsearch=Room&rhhlterm=Room#t=Drag_And_Drop%2FDrag_And_Drop_Overview%2FDnD_Overview.htm&ux=search

YoYo Games Ltd. (2022c), GameMaker Manual Creating Sprites. Haettu 20.11.2022 osoitteesta https://manual.yoyogames.com/Quick_Start_Guide/Creating_Sprites.htm

YoYo Games Ltd. (2022d), GameMaker Manual Creating Sound Effects. Haettu 21.11.2022 osoitteesta https://manual.yoyogames.com/Quick_Start_Guide/Creating_Sound_Effects.htm

YoYo Games Ltd. (2022e), GameMaker Manual Rooms. Haettu 21.11.2022 osoitteesta https://manual.yoyogames.com/Quick_Start_Guide/Rooms.htm

YoYo Games Ltd. (2022f), GameMaker Manual Objects And Instances. Haettu 21.11.2022 osoitteesta https://manual.yoyogames.com/index.htm#t=Quick_Start_Guide%2FObjects_And_Instances.htm

YoYo Games Ltd. (2022g), GameMaker Manual Object Events. Haettu 25.11.2022 osoitteesta https://manual.yoyogames.com/The_Asset_Editors/Object_Properties/Object_Events.htm

YoYo Games Ltd. (2022h), GameMaker Manual Networking. Haettu 27.11.2022 osoitteesta https://manual.yoyogames.com/The_Asset_Editors/Object_Properties/Async_Events/Networking.htm

YoYo Games Ltd. (2022i), GameMaker Manual Networking. Haettu 27.11.2022 osoitteesta https://manual.yoyogames.com/Additional_Information/Guide_To_Using_Buffers.htm

YoYo Games Ltd. (2022j), GameMaker Manual GML Reference Networking. Haettu 11.11.2022 osoitteesta https://manual.yoyogames.com/GameMaker_Language/GML_Reference/Networking/Networking.htm

YoYo Games Ltd. (2022k), GameMaker The Ultimate 2D Game Engine. Haettu 1.12.2022 osoitteesta https://gamemaker.io/en

The Open Group (2018), The Open Group Base Specifications Issue 7 IEEE Std 1003.1-2017. Haettu 1.12.2022 osoitteesta https://pubs.opengroup.org/onlinepubs/9699919799/functions/socket.html

Postel, J. (1983). RFC 879: The TCP Maximum Segment Size and Related Topics. Haettu 10.11.2022 osoitteesta https://datatracker.ietf.org/doc/html/rfc879#section-5

Verma, S. (2022), What is TCP 3 Way Handshake? Haettu 14.10.2022 osoitteesta https://www.scaler.com/topics/computer-network/tcp-3-way-handshake/

Imperva Inc. (2022), Used datagram protocol (UDP). Haettu 15.10.2022 osoitteesta https://www.imperva.com/learn/ddos/udp-user-datagram-protocol/

The Center for Digital Games. (2022), YoYo Games. Haettu 17.10.2022 osoitteesta https://www.cdgr.ucsb.edu/database/organization/1764

Caroll, M (2022), PubNub UDP vs TCP: Why to Run Gaming Services Seperate from Chat. Haettu 20.10.2022 osoitteesta https://www.pubnub.com/blog/why-you-should-run-your-game-servers-separate-from-your-chat/

Datapath.io (2017), The History of Online Gaming. Haettu 07.12.2022 osoitteesta https://medium.com/@datapath_io/the-history-of-online-gaming-2e70d51ab437