Ohjelmoinnin prototyyppi

Sisällysluettelo

Sisällys

[1. Johdanto 3](#_Toc19604120)

[2. Ohjelman teossa käytetyt välineet 4](#_Toc19604121)

[3. Ohjelman UML-mallinnus 5](#_Toc19604122)

[4. Ohjelman teon vaiheet 6-27](#_Toc19604123)

[5. Ohjelman testaus ja versionhallinta 27](#_Toc19604124)

[6. Tietokanta 28](#_Toc19604125)-33

[7. Loppupäätelmät 34](#_Toc19604126)

[Lähteet 35](#_Toc19604127)

# 1. Johdanto

Näyttö suoritetaan luomalla 2-uloitteinen Endless Runner- peli. Pelin tarkoituksena on päästä mahdollisimman pitkälle osumatta vastaan tuleviin esteisiin. Pelin pisteet lasketaan ohitettujen esteitten lukumäärän mukaan.

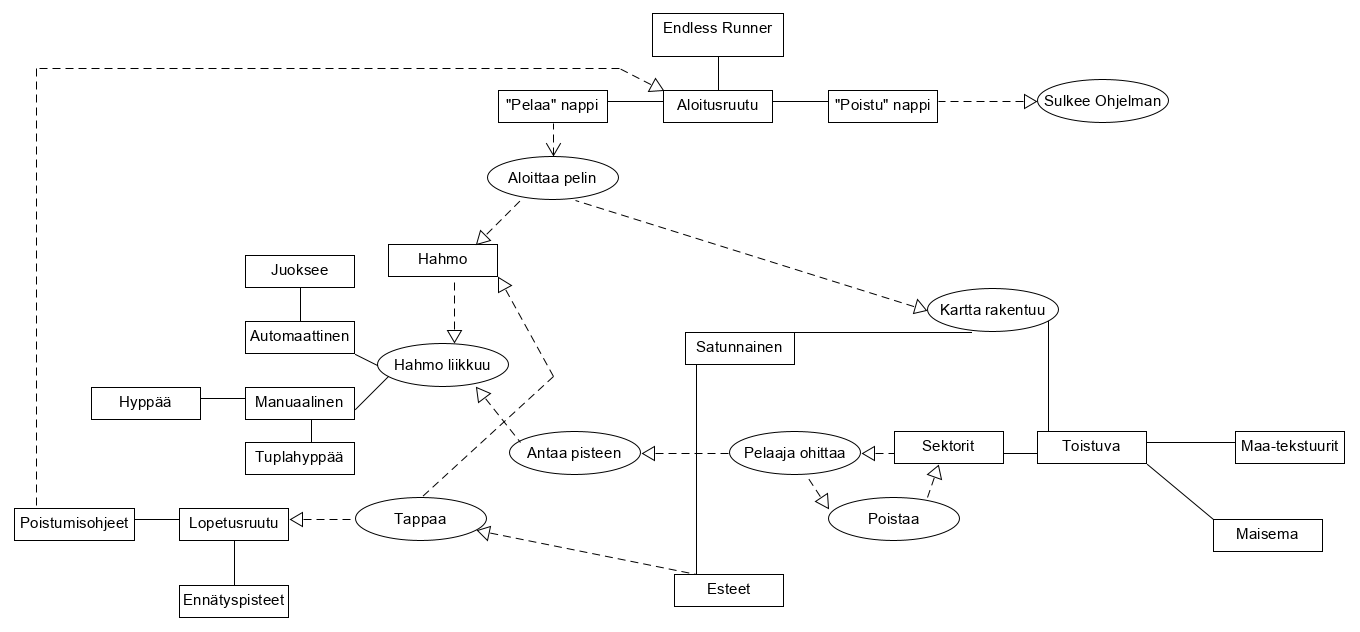
# 2. Ohjelman teossa käytetyt välineet

Ohjelmointi: Visual Studio Code

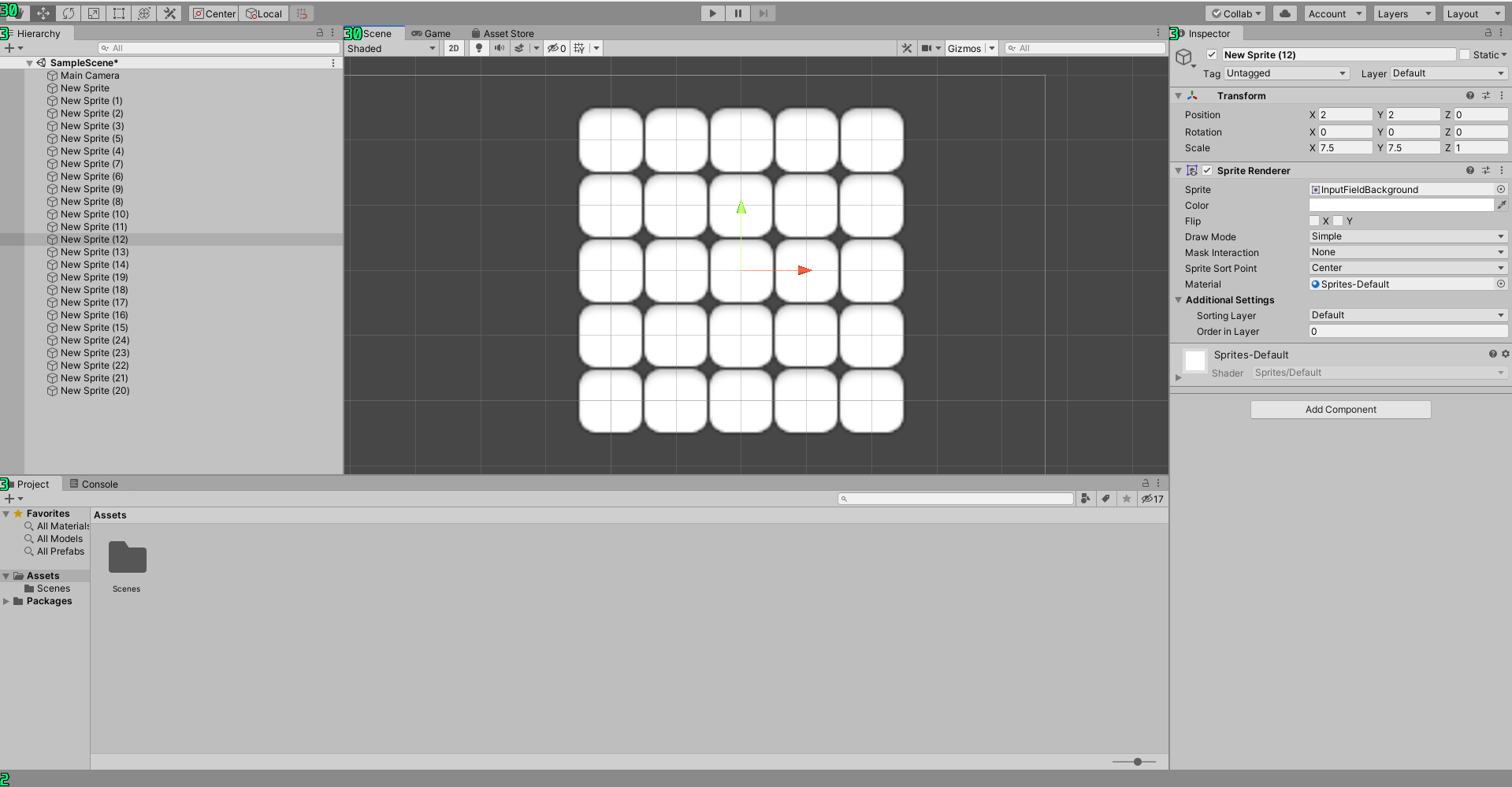
Pelimoottori: Unity

UML-mallinnus: UMLet

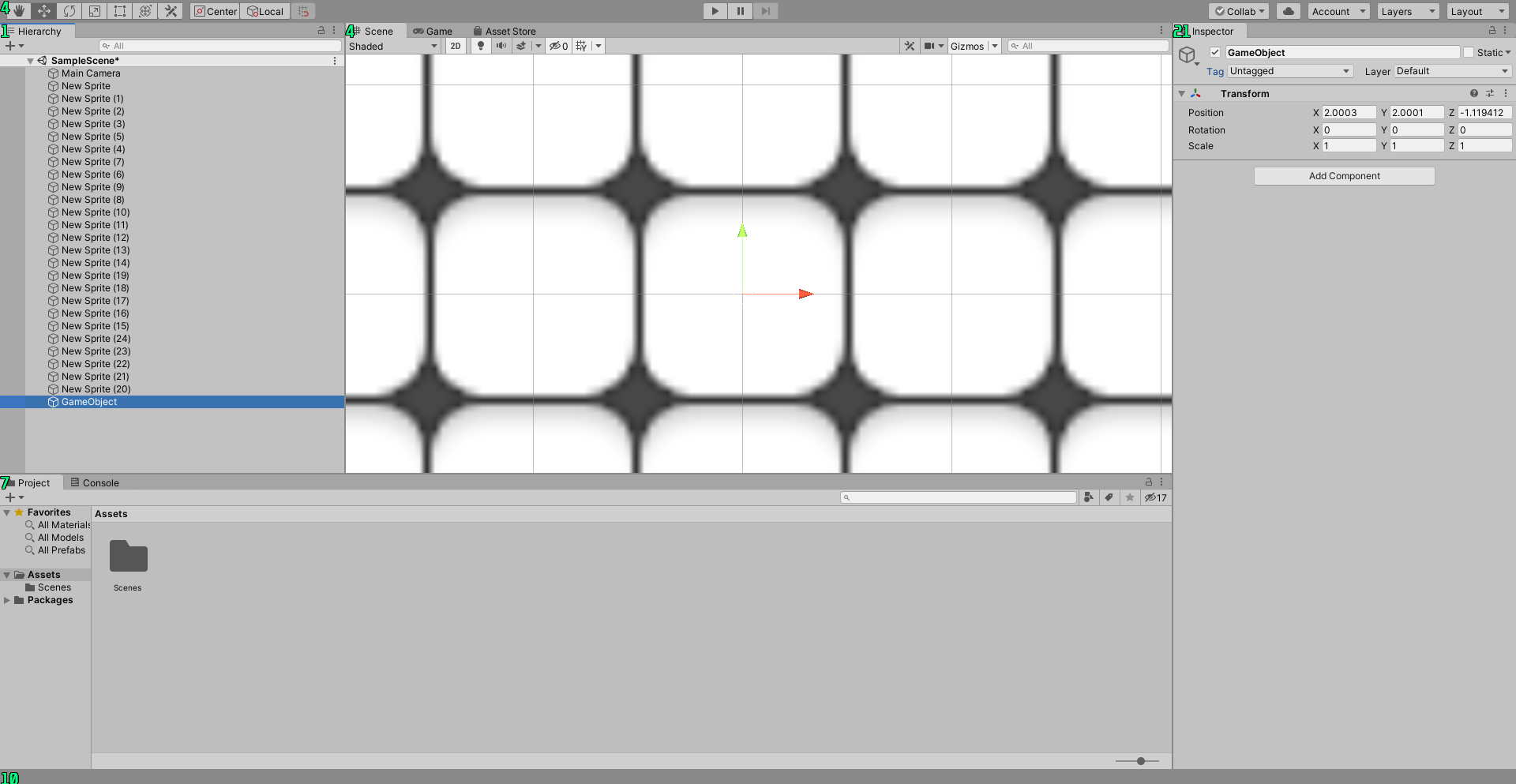
# 3. Ohjelman UML-mallinnus



# 4. Ohjelman teon vaiheet



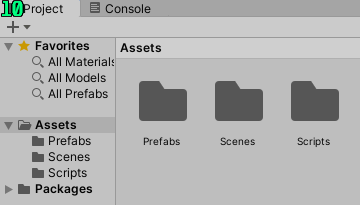
Tehdään 2D projektiin kartan yksittäisen osion runko sprite objekteista.



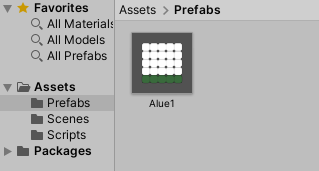
Annetaan alueelle kiintopiste jota käyttäen se generoituu oikeaan kohtaan.



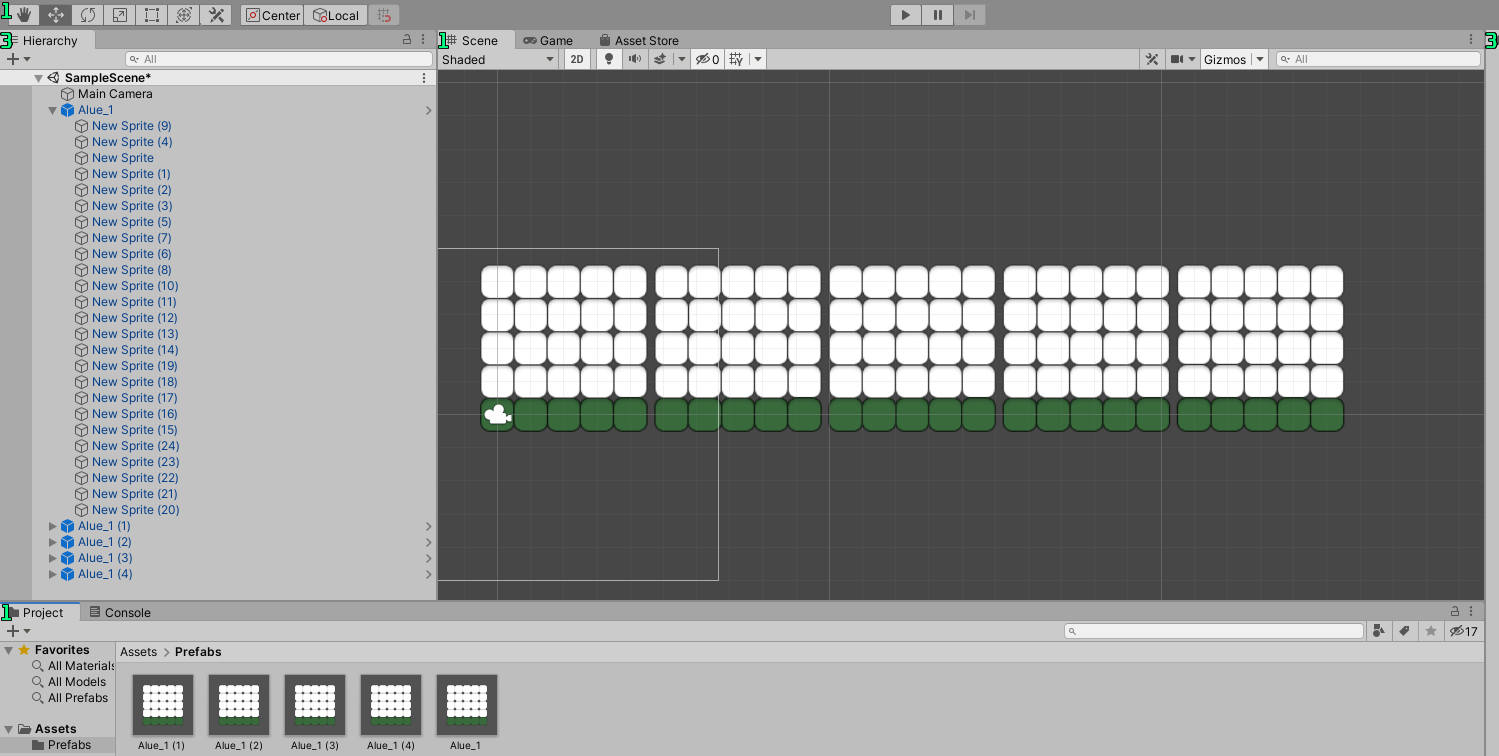
Siirretään tämän alueen runko kiintopisteen alle.



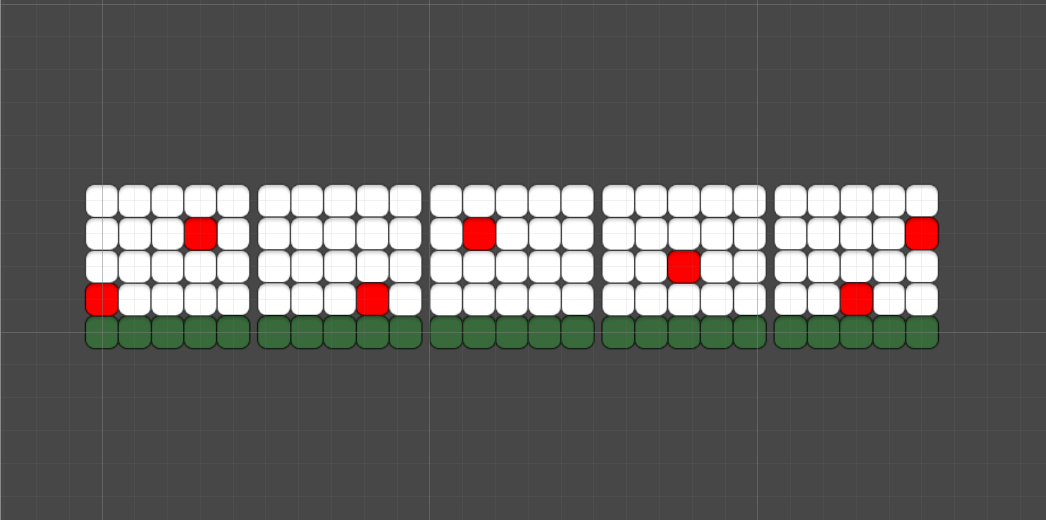
Luodaan uusi kansio elementeille ja skripteille



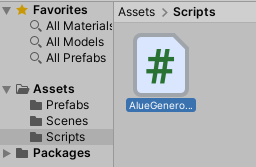
Siirretään alue elementteihin. tämä helpottaa pelin kentän generoimista ja hallintaa.



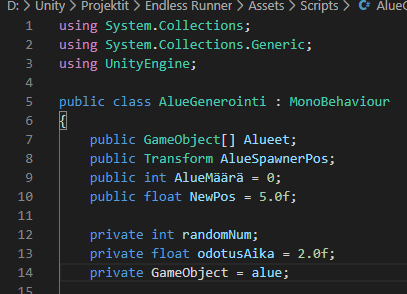
kopioidaan alueita muutama ja siirretään ne myös elementteihin



määritetään alusta ja tulevat esteet



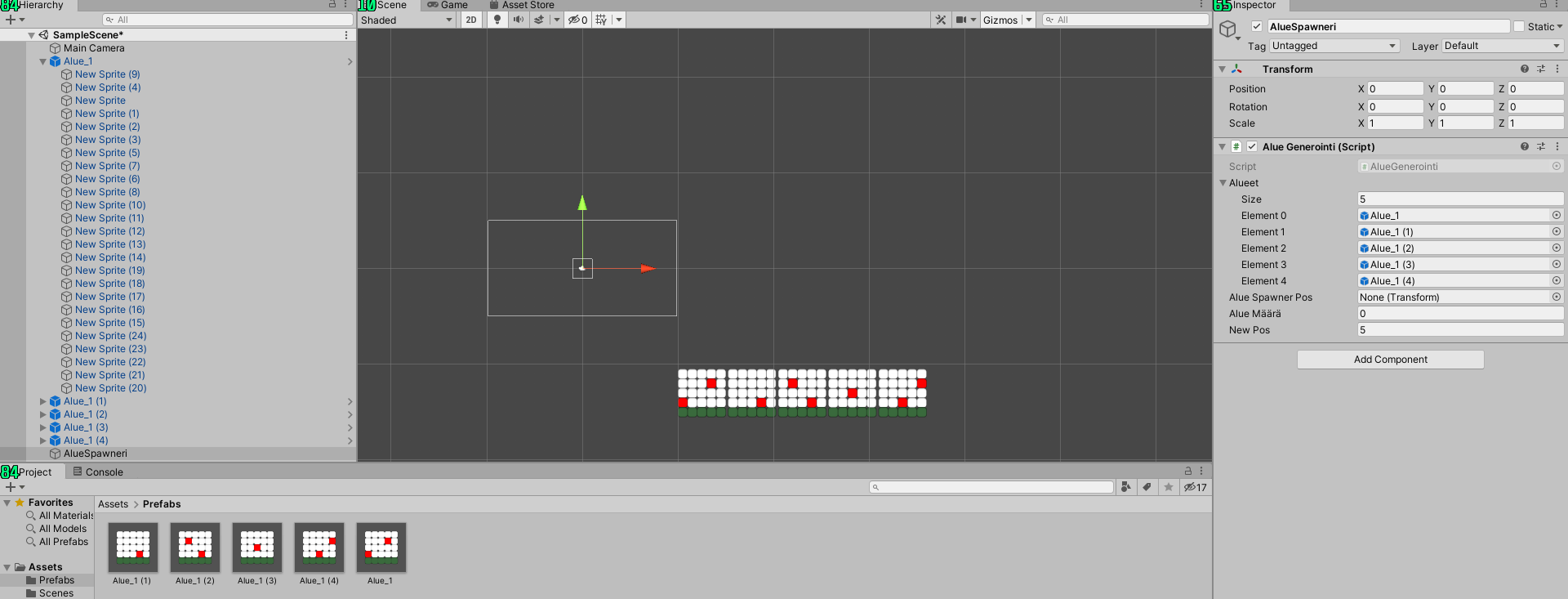
tehdään skripti jolla generoidaan alueita



määritetään skriptissä tarvittavat muuttujat



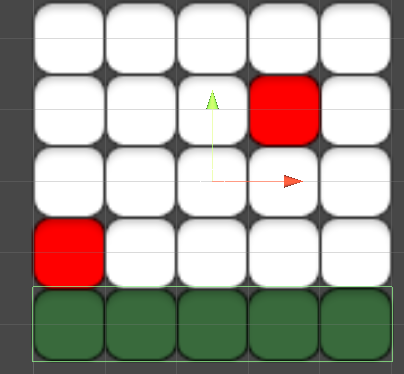
loppu koodi kommentoituna



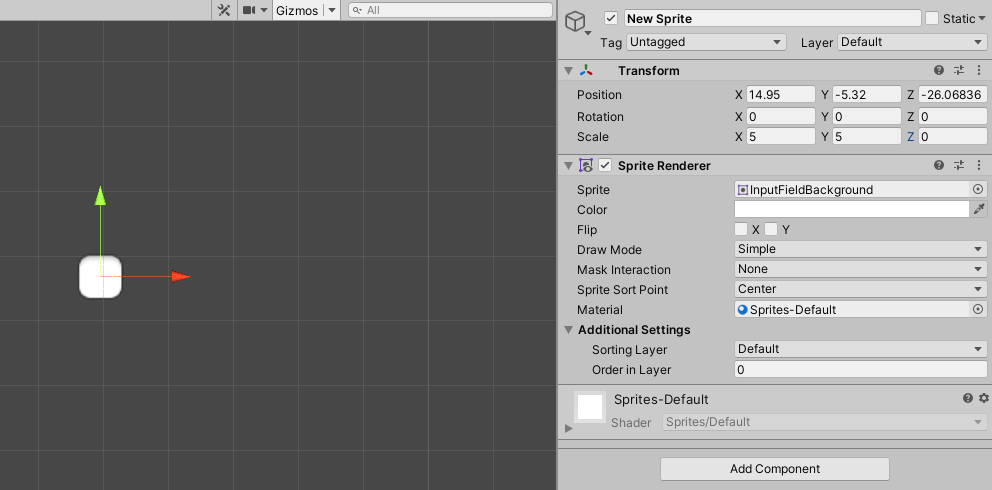
Tehdään piste josta alueiden generointi alkaa



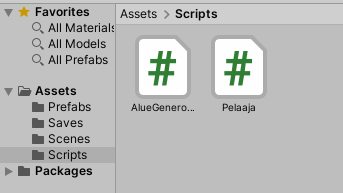
kun suoritetaan ohjelma, se luo satunnaisesti alueita peräjälkeen



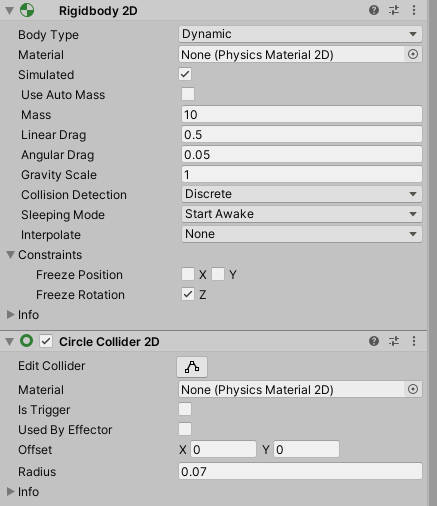
luodaan maa osiolle hitboxit, tehdään tämä kaikille alueille



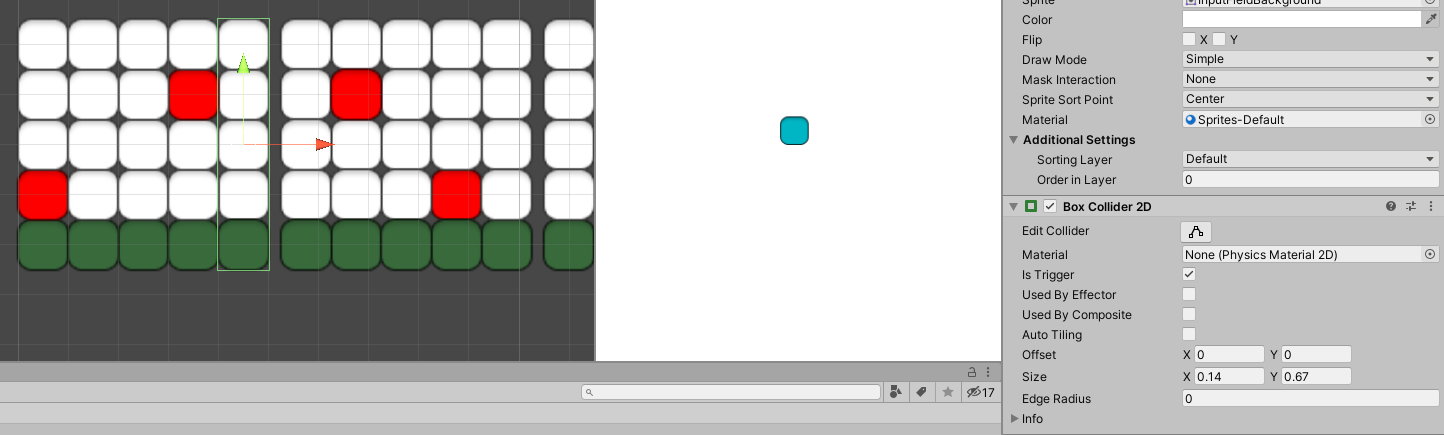
tehdään pelaaja objekti



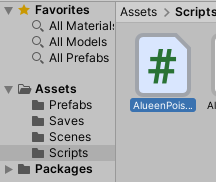
Tehdään pelaajalle skripti



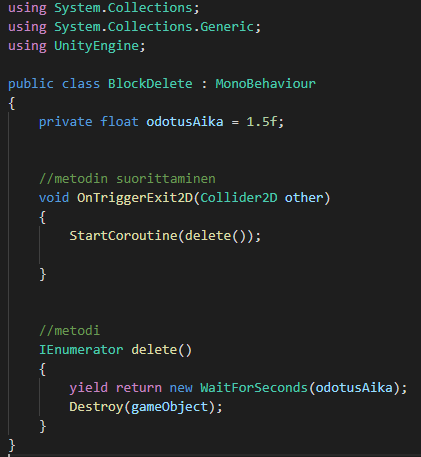
tehdään pelaajalle rigidbody ja hitbox komponentit



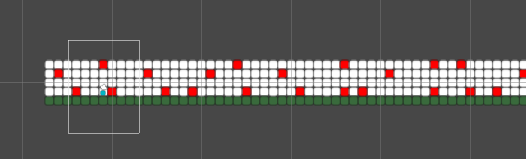
annetaan alueille pisteet jotka ohittamalla, tietyn matkan päästä entiset poistuu



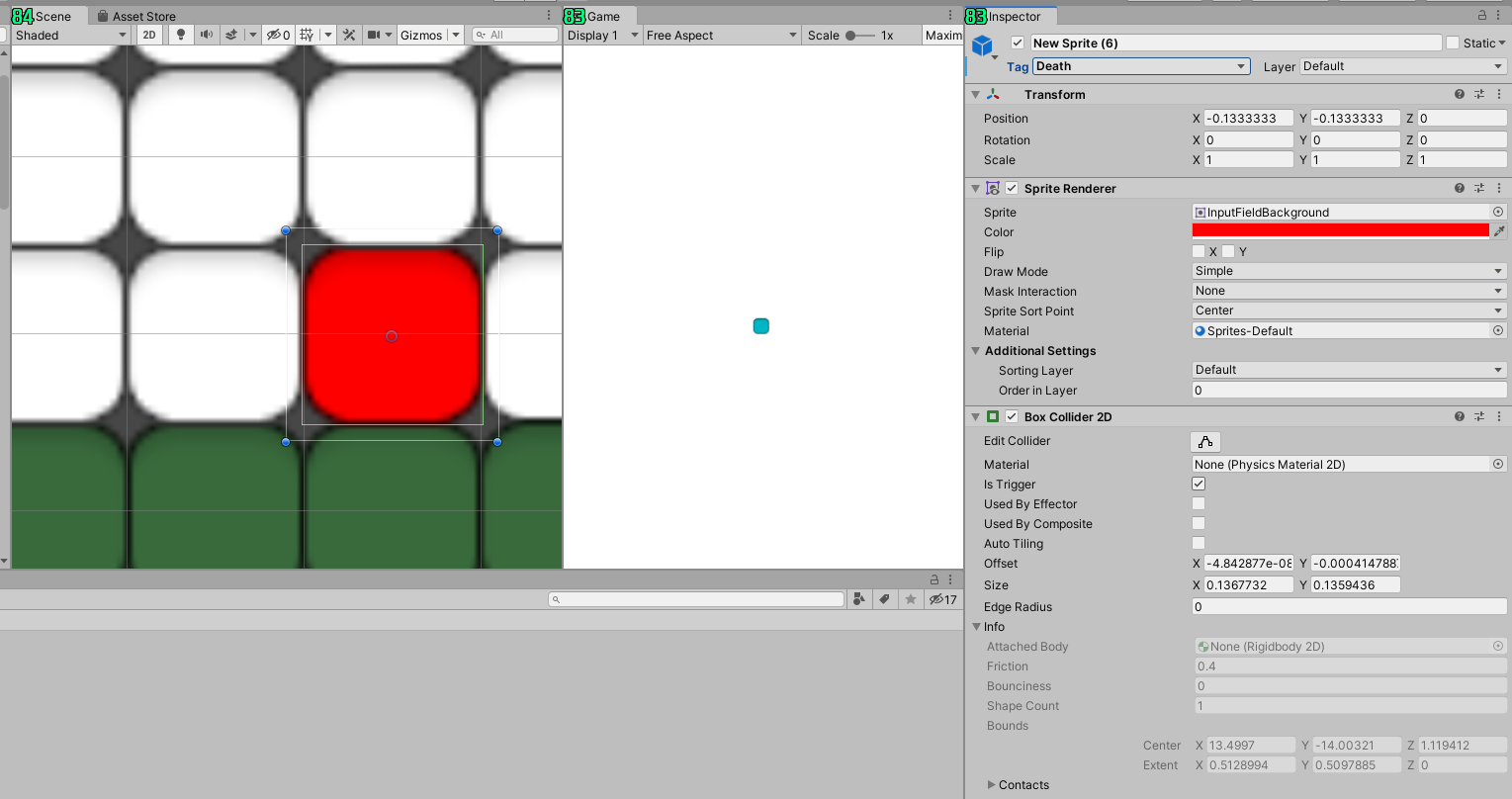
tehdään tälle skripti



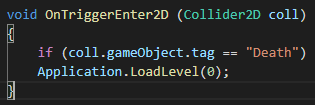
skriptin sisältö kommentoituna



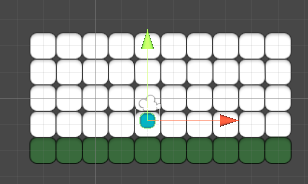
lopputulemana edelliset alueet poistuvat muistin vapauttamiseksi



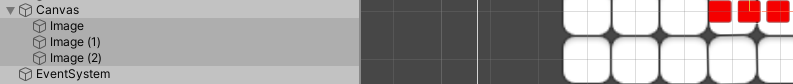
annetaan esteille hitboxit ja tägi, jolla hallitsemme sen reagointia pelaajaan



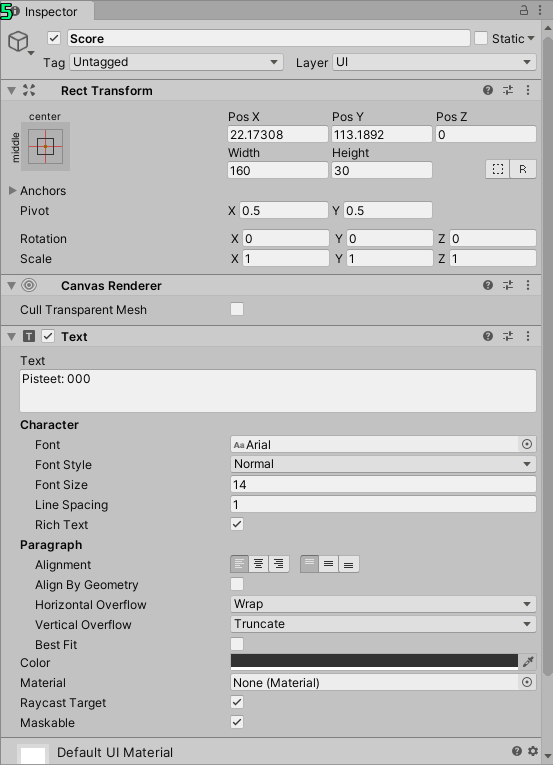
määritetään koodiin ”Death” tägin tarkoitus, resettaa siis pelin eli pelaaja kuolee



tehdään myös pelaajalle mukava aloitus jotta se ei putoa suoraan esteen päälle pelin alkaessa



Elämiä varten tehdään Canvas johon kolme kuvaelementtiä



tehdään pisteille teksti elementti



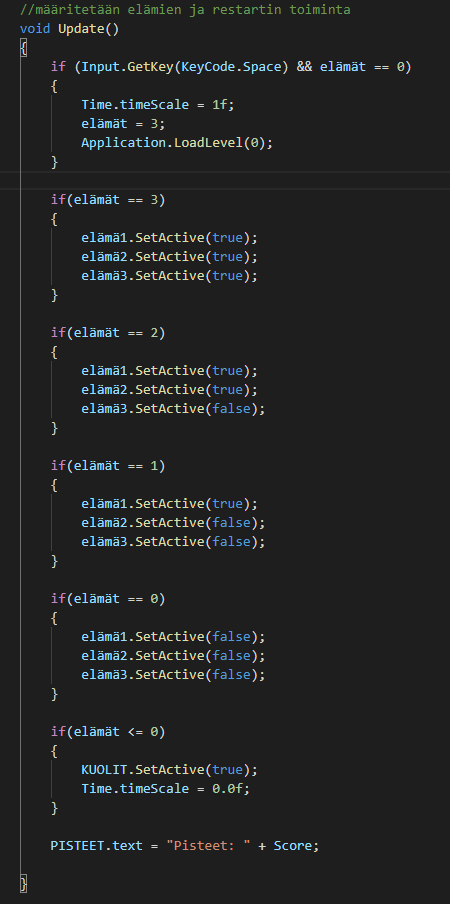
erilaisia fontteja varten voidaan luoda niille oma kansio



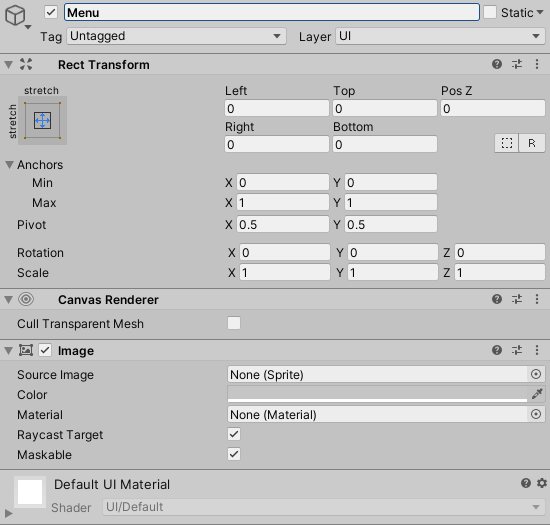
luodaan näillä lopetusruutu



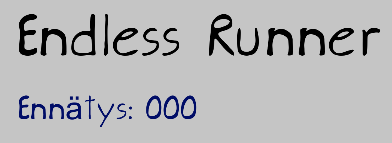
Pari muutosta kuolemaan ja pisteenlaskun toiminta



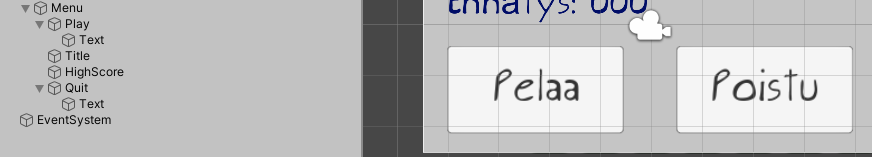
Elämien ja restartin toiminta



Tehdään Menu paneeli



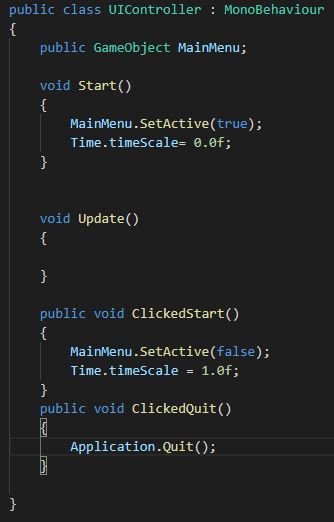
Tehdään menuun tekstit



Tehdään menuun napit



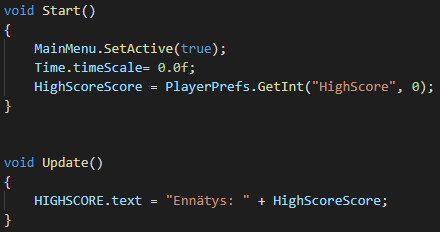
uusi skripti käyttöliittymän ohjaamiseen

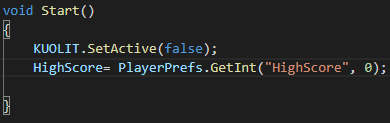


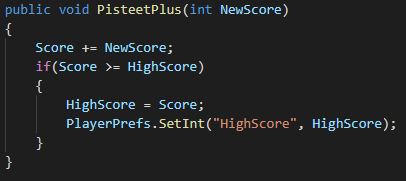
perus skripti napeille



nyt voidaan tehdä toimiva buildi jossa napit toimivat



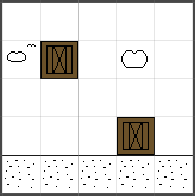




ennätyksien säilyttämiseksi muutoksia skripteihin



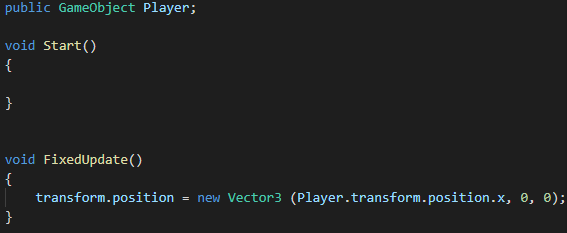
tehdään muutoksia jolla luodaan tuplahyppy



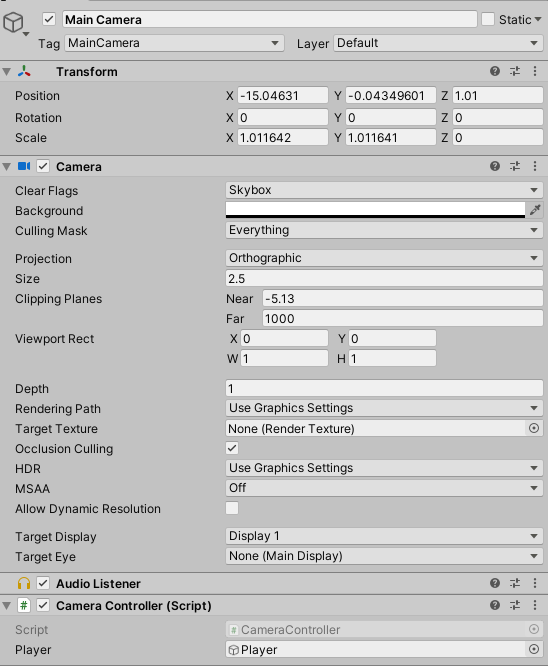
tehdään taustalle, maalle ja esteille tekstuurit



tehdään kameran seuraamisen skripti



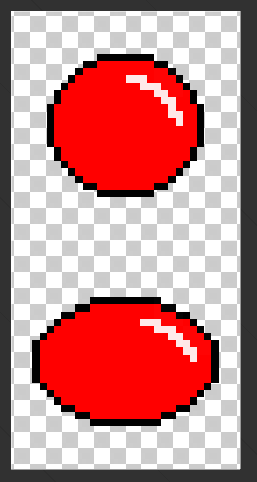
Skriptissä määritetään kameran liike



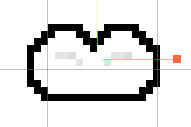
Main Camera irroitetaan pelaajasta ja lisätään siihen skripti



tehdään linnulle pieni animaatio



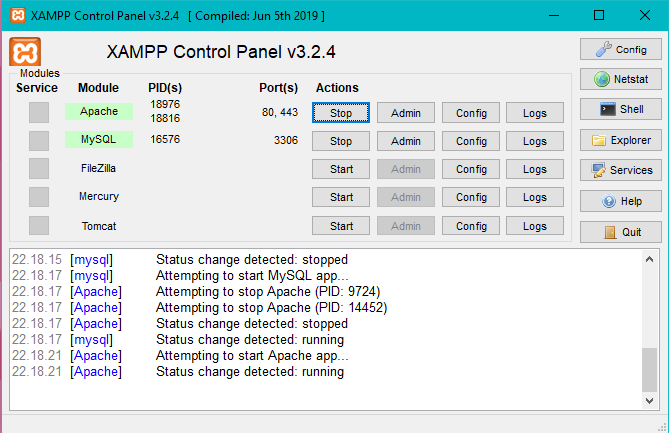
Myös pelaajalle tehdään ulkonäkö, että myös animaatio



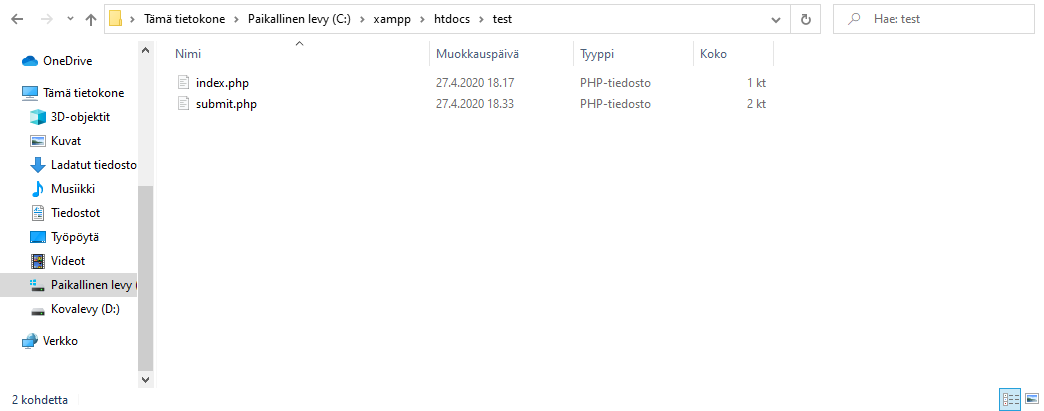
Uudistetaan pilven ulkonäköä, ja tehdään sille hiukan elävämpi animaatio

# 5. Ohjelman testaus ja versionhallinta

# 6. Tietokanta



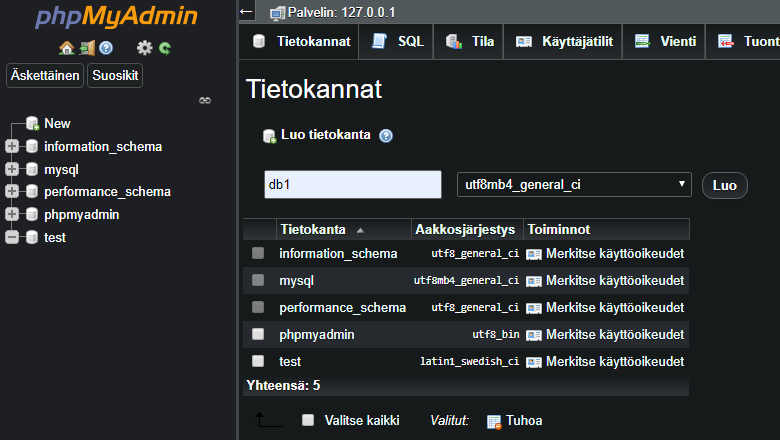
XAMPP:in aloitusnäkymästä kytketään halutut moduulit käyntiin, tässä tapauksessa tehdään MySQL tietokanta Apachen web palvelinta käyttäen.



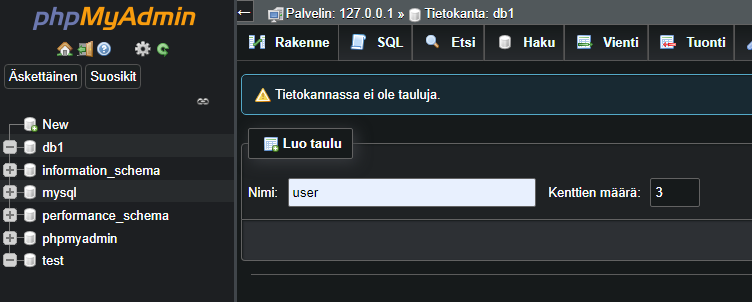
XAMPP:in tiedostosijainnista löytyvän htdocs kansion alle luodaan kansio esimerkkinimellä ”testi”. Testikansioon luodaan kaksi eri tekstitiedostoa, index.php ja submit.php.



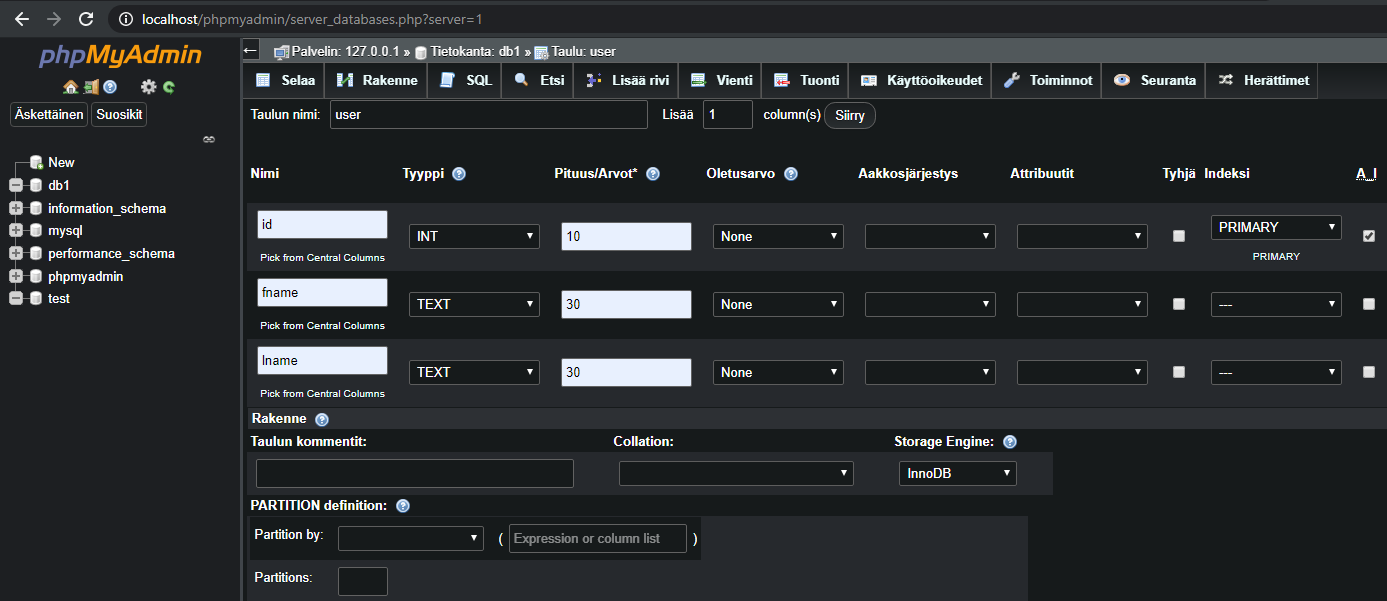
Kirjoittamalla selaimeen ”localhost”, päästään Apache web palvelimeen käsiksi. Seuraavaksi painetaan oikeasta ylänurkasta löytyvää ”phpMyAdmin” painiketta.



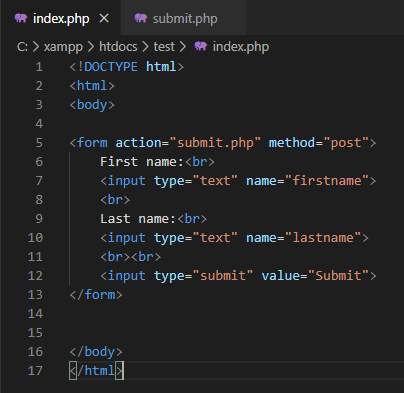
Luodaan uusi tietokanta nimeltä ”db1”.



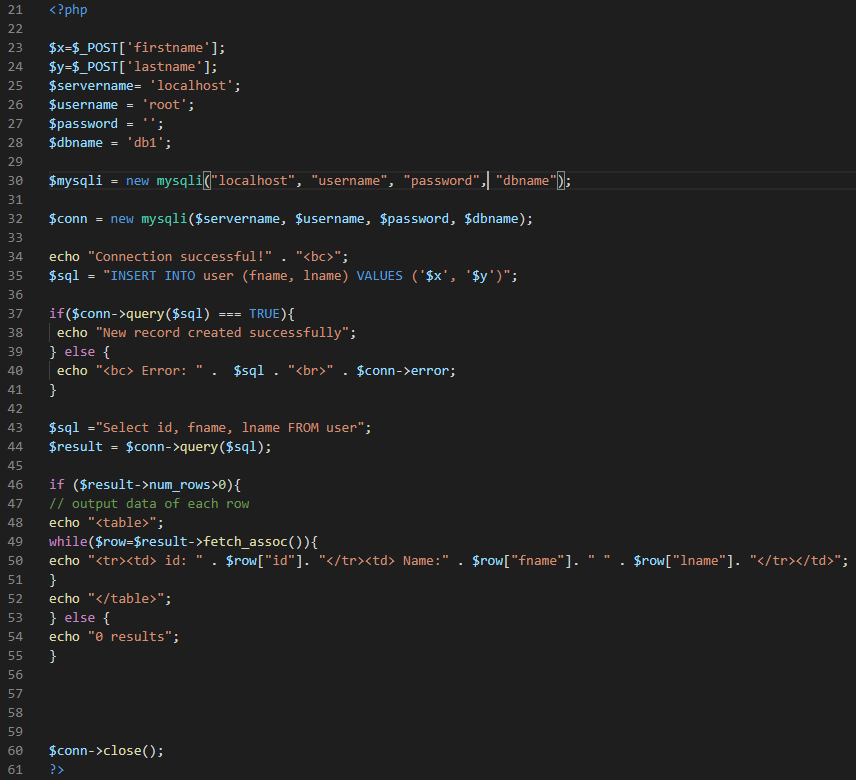
Sitten uudelle tietokannalle taulu nimeltä ”user”, jolle annettaan kolme eri arvollista kenttää.



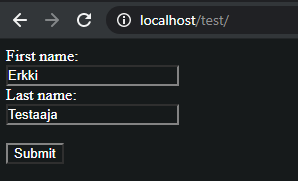
”id”, ”fname”, ja ”lname” kentät, joille määritetään minkälaisia arvoja niiden sisään pystytään syöttämään.



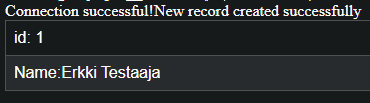
Index.php koodissa määritetään nettisivun kyselykenttä, ja sen kirjoitus ja lähetys palkit.



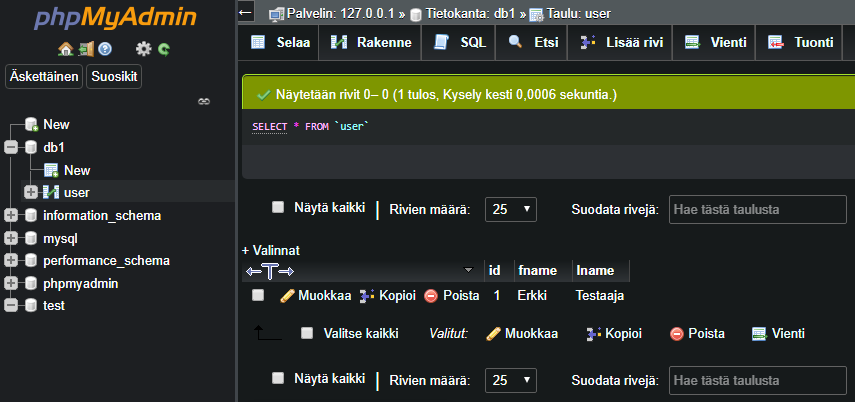
Submit.php koodissa kyselyssä lähetetyt arvot sijoitetaan luomaamme tietokantaan niille kuuluviin kenttiin.



Kokeilemme lisätä ”Erkki Testaaja” nimisen käyttäjän tietokantaan.



Ilmoitusruusun mukaan kysely on suoriutunut onnistuneesti.



Tietokantaan on ilmestynyt ”Erkki Testaaja” niminen sarake.

# 7. Loppupäätelmät

Projekti onnistui mielestäni ihan hyvin. Alkuvaiheessa kohdatut ongelmat korjattiin järjestelmällisesti saavuttaen täysin toimivan ja virheettömän pelin. Joitain hienosäädön kohtia pelistä saattaa toki löytyä, mutta mitään pelikokemusta heikentävää tai pilaavaa ei ohjelman sisään jäänyt. Ulkoisesti pelistä tuli omasta mielestäni myös täysin siedettävän ja yksinkertaisen näköinen, eikä ylimääräisten painikkeiden taakse paljoa pysty eksymään. Word dokumentin puoli onnistui myös odotetusten mukaan. Dokumentointi ohjelman tekovaiheessa ei tuottanut sen suurempia ongelmia taikka päänvaivaa. XAMPP:in dokumentointi eteni myös moitteettomasti.

# Lähteet

Koodilähteitä:

<https://www.w3schools.com/>

<https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/>

Muistaakseni käytin näitä videoita apuna alueiden generoinnin toteuttamisessa:

<https://www.youtube.com/watch?v=_D_GjO81Oyg>

<https://www.youtube.com/watch?v=0SJgb0XrwvA&t=1063s>