

Y2 projekti

1. Henkilötiedot:

Lauri Paloneva, 908445, Kauppakorkeakoulu (tieto- ja palvelujohtamisen pääaine), 2. vuosikurssi

2. Yleiskuvaus:

Tarkoituksen tehdä vuoropohjainen strategiapeli, jossa pelaajalla, sekä tekoäly vastustajalla on 4x4 kokoinen ruudukko pelialustana. Molemmilla osapuolilla on ruudukolla 1–3 hahmoa. Vuorolla pelaaja voi päättää haluaako hän tehdä iskun vai liikuttaa jotain hahmoista. Vuorollaan voi ainoastaan tehdä yhden iskun tai yhden siirron.

Pelin tavoitteena on tuhota vastustajan hahmot, ennen kuin vastustaja tuhoaa omat hahmot.

Iskut tekevät eri määrän vahinkoa hahmoihin, sekä osuvat eri alueelle 4x4 ruudukkoa. Esimerkiksi yksi isku voi tehdä 2 elämäpistettä vahinkoa ja vaikuttaa 2x2 kokoiselle alueelle. Toinen isku voi tehdä 5 elämäpistettä vahinkoa, mutta vaikuttaa vain yhteen ruutuun.

Hahmoilla on nopeuselementti, joka määrittää kumman pelaajan isku tapahtuu ensin. Tämän lisäksi hahmoilla on eri elämäpisteitä, jotka määrittävät montako iskua hahmo kestää. Jos pelaaja siirtää hahmoa tämä tapahtuu aina ennen iskua, vaikka hahmo olisikin hitaampi.

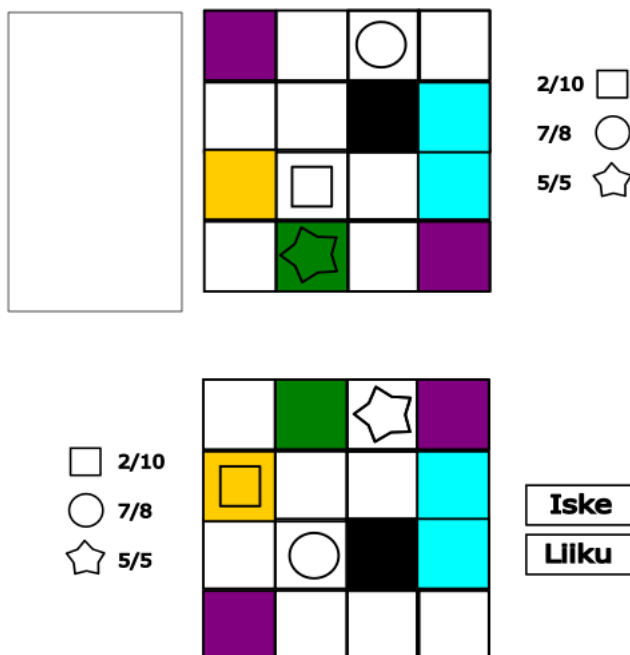
Hahmoihin voi kohdistua erilaisia vaikutuksia. Näitä voivat aiheuttaa Isku tai pelialustan laatta. Esimerkiksi Myrkkylaatta aiheuttaa hahmolle myrkytyksen, joka aiheuttaa vahinkoa muutaman vuoron ajan. Muita vaikutuksia on esimerkiksi liikkumisen estyminen vuoron ajaksi tai laatalle liikkumisen estyminen.

Grafiikkana toimii 2 kpl 4x4 pelialustoja. Hahmot ovat erilaisia muotoja, kuten ympyröitä ja kolmioita.

Tekoäly vastustaja noudattaa 3 eri strategiaa. Alustavasti olen ajatellut suorittaa projektin vaativana versiona.

3. Käyttötapakuvaus ja käyttöliittymän luonnos:

Kun pelaaja avaa ohjelman, on tarkoitus esiintyä simppeli käyttöliittymä, missä kysytään pelaajalta, haluaako hän jatkaa edellistä peliä vai aloittaa alusta. Pelaajalla on mahdollisuus valita tekoälyn strategia. Jos pelaaja valitsee

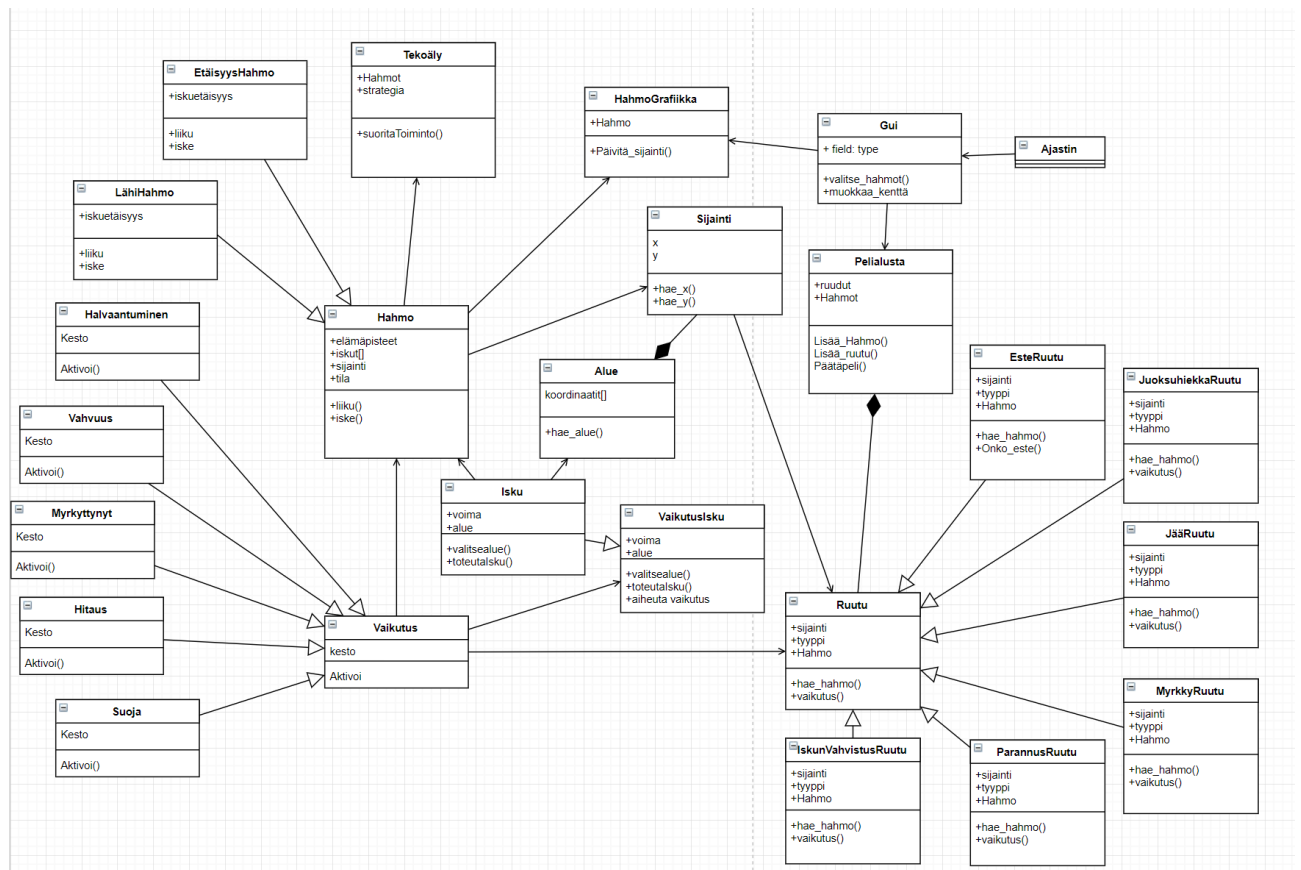


Pelaajan on tehtävä päätös vuorollansa suoritettavasta toiminnasta. Painamalla ”Iske” nappia pelaaja valitsee iskevänsä ja toisinpäin. Valittuaan Iskun tai liikkumisen, pelaaja valitsee hiirellään ruudun, johon hän iskee tai liikkuu. Värilliset laatat kuvaavat erilaattojen ominaisuuksia. Tekstilaatikko kertoo mahdollisista vaikutuksista.

Käyttökuvaus:

Pelaaja avaa ohjelman, ja voi valita haluaako hän muokata kenttää vai pelata. Valitessaan pelaamisen häneltä kysytään, haluaako hän jatkaa edellistä peliä vai aloittaa alusta. Kun pelaaja aloittaa pelin on käyttöliittymä yllä olevan kuvan näköinen. Kun Pelaaja painaa iske, hän saa valita hahmon, jolla haluaa iskeä. Kun hahmo on valittu, pelaaja valitsee mihin kohtaa vastustajan ruudukkoa hän haluaa iskeä. Tämän jälkeen isku ja tekoälyn isku toteutuvat hahmojen nopeuden määrittämässä järjestyksessä. Mikäli iskut osuvat ne aiheuttava vahinkoa ja mahdollisen vaikutuksen hahmoon.

4. Ohjelman rakennussuunnitelma



Yllä UML luokkadiagrammi, joka hahmottelee Ohjelman toimintaa. UML diagrammista on nähtävillä Luokkien tärkeimpiä metodeja.

Hahmo luokka kuvaa hahmoa, sen perivät EtäisyysHahmo ja lähihahmo, jotka iskevät eri etäisyyksille. Hahmo luokka luo kaiken hahmoon kuuluvan toiminnallisuuden. Hahmoja voidaan kustomoida pitkälti Hahmoluokan elämäpiste, nopeus ja iskut attribuuteilla, joten useamman Hahmoalaluokan luominen ei ole tarpeellista. Hahmo luokan tärkeimmät metodit ovat isku, joka toteuttaa iskun, sekä liiku, joka toteuttaa hahmon siirron.

Pelialusta koostuu Ruutu luokan olioista. Pelialusta luokka luo pelialustan ja sijoittaa sinne hahmot, sekä ruudut.

Ruutuluoalla useita UML kaavioissa olevia alaluokkia. Ruutu luokka käyttää Vaikutus luokkaa, sillä sen alaluokat aiheuttavat jonkin vaikutuksen. Vaikutusluokalla on useita alaluokkia.

Sijainti luokka määrittää asioiden sijainnin. Hahmoilla ja Ruutuilla on oltava sijainti, jotta tiedetään osuvatko iskut ja vaikuttavatko ruudut.

Alue luokka koostuu useista sijainneista. Sitä käytetään isku luokassa, kun osa iskuista osuu useampaan koordinaattiin kerrallaan.

Isku luokassa luodaan Iskun toiminnallisuus, eli voima ja alue. Sen alaluokka vaikutusisku aiheuttaa lisäksi jonkin vaikutuksen hahmoon.

Hahmografiikkaluokassa luodaan Hahmojen graaffiset ominaisuudet, sekä liikutetaan elementtejä, jos ne siirtyvät.

Gui luokassa luodaan käyttöliittymään liittyvät elementit, kuten napit ja niiden toiminnallisuudet. Esimerkiksi muokkaa_kenttää metodi.

5. Tietorakenteet

Tietorakenteet, mitkä sopivat parhaiten ohjelmaan on listat ja sanakirjat. Esimerkiksi hahmojen eri ominaisuudet on hyvä tallentaa sanakirjaan, jolloin hahmon nimeä voidaan käyttää avaimena. Pelikentällä olevia hahmoja yms. on järkevä tallentaa listaan. Dynaamiset rakenteet ovat tarpeellisia, kun esim. käytetään listaa hyväksi määrittelemään pelikentällä olevat hahmot. Tällöin on tärkeää, että pystytään poistamaan hahmoja kentältä.

6. Tiedostot ja tiedostoformaatit

Ohjelman on tarkoitus käsitellä tekstitiedostoja. Hahmot ja niiden ominaisuudet on tarkoitus tallentaa tekstinä niin, että eri ominaisuudet on erotettu jollain välimerkillä.

Esimerkiksi csv tiedosto on tähän tarkoitukseen hyvä. Samalla tavalla on myös järkevä tallentaa eri laattojen ominaisuudet.

7. Algoritmit

Tekoälyn algoritmi:

Tarkoituksena luoda esimerkiksi passiivinen, aggressiivinen ja odotusarvo strategia tekoälylle.

Passiivinen strategia pyrkii pitämään omia hahmoja hengissä. Algoritmi liikuttaa hahmoja, mikäli niihin on edellisellä vuorolla tehty isku, jonka voisi välttää liikkumalla. Jos pelaaja ei ole osunut edellisellä vuorolla, tekoäly pyrkii tekemään iskun, joka tuottaisi eniten kokonaisvahinkoa. Algoritmi ei osaa ennustaa pelaajan siirtoa.

Aggressiivinen strategia pyrkii tekemään joka vuoro eniten kokonaisvahinkoa tuottavan iskun. Se liikuttaa omaa hahmoaan ainoastaan, jos hahmo tuhoutuu seuraavalla vuorolla iskusta. Algoritmi ei osaa ennustaa pelaajan siirtoa.

Odotusarvo strategia pyrkii tekemään iskun, mikä on odotusarvoltaan suurin. Algoritmi laskee pelaajan hahmojen mahdolliset liikkeet, ja päättelee näiden perusteella todennäköisyyden iskun osumiselle. Tämän jälkeen se määrittää iskun odotusarvon kertomalla osumistodennäköisyyden iskun vahinko määrällä. Strategia suorittaa iskun, jolla on suurin odotusarvo. Strategia olettaa, että pelaaja tekee aina iskun, mikä tuottaa eniten vahinkoa. Strategia laskee todennäköisyyden tämän

osumiselle, mikäli tekoäly itse siirtää hahmoaan ja laskee odotusarvon vältetyille vahingolle. Mikäli vältetyn vahingon odotusarvo on suurempi, kuin minkään toteutettavan iskun odotusarvo, suorittaa strategia hahmon siirron.

Satunnaistapahtumien laskemista varten hyödynnetään Pythonin Random algoritmia.

8. Testaussuunnitelma

Luettaessa tiedostoista hahmojen ja laattojen ominaisuuksia, pitää testata onko tiedot luettu oikein ja onko ominaisuudet oikeassa formaatissa.

Käyttöliittymää ja pelialustaa tehtäessä pitää testata, että piirretty alusta on halutun lainen.

Pelin ollessa käynnissä, pitää testaamaan, toimiiko hahmojen siirtäminen oikein, ja aiheuttaako iskut oikean vaikutuksen oikeisiin hahmoihin.

9. Kirjastot

Kirjaston käytetään PyQt6 kirjastoa käyttöliittymän luomiseen.

10. Aikataulu

Tarkoituksena on aloittaa luomalla luokat Ruutu, Pelialusta ja Gui. Tämän jälkeen lähdetään toteuttamaan Hahmo ja sijainti luokkaa. Tarkoituksena on saada eräänlainen palikkaversio pelistä aluksi, jossa ei ole vielä Hahmojen tai laattojen erilaisuutta tai suurempaa graafista yksityiskohtaa. Näin voidaan testata pelin perusominaisuuksia, kuten iskemistä ja liikettä vähemmällä työmäärällä. Tämän jälkeen voidaan ruveta lisäämään peliin eri ominaisuuksia ja kustomointeja. Viimeisenä toteutettavat luokat ovat Hahmo, Vaikutus ja Ruutu luokkien alaluokat.

27.2–5.3 välisenä aikana aion suorittaa 5. ja 6. kierrosten tehtävät loppuun.

1. välipalautus

5.3–24.3 välisenä aikana on tarkoitus olla luotuna käyttöliittymä, jolla voi valita pelin toiminnallisuuksia. Tarkoituksena on pystyä aloittamaan peli, niin että se piirtää pelialustan. Kuitenkaan mitään siirtoihin liittyvää toiminnallisuutta ei välttämättä ole valmiina tähän mennessä. Tähän arvioin käyttäväni noin 15 h aikaa.

2. Välipalautus

24.3–14.4 välisen aikana on tarkoitus toteuttaa Hahmoon liittyvä toiminnallisuus. Tässä vaiheessa on tarkoitus pystyä avaamaan peli ja tekemään iskuja ja siirtoja käyttöliittymän avulla. Tässä vaiheessa Tekoäly ei välttämättä toteuta hienostunutta tai lopullista strategiaa.

Tässä vaiheessa on tarkoitus olla luotuna tiedostojen lukemiseen ja pelialustan piirtoon liittyvät testit. Tähän arvioin käyttäväni noin 15–20 h aikaa.

Lopullinen palautus

14.4–12.5 välisen aikana aion suorittaa loput peli toiminnallisuudesta. Tähän kuuluu alaluokkien teko ja tekoälyn algoritmin teko. Lisäksi muu pelin loppuun hiominen. Tähän arvioin kuluvan 20-25h aikaa.

11. Kirjallisuusviitteet ja linkit

Python Software Foundation. (2021). Python 3 documentation. Retrieved from

<https://docs.python.org/3/>

