

Tekninen suunnitelma

Ville Strengell 594590

04.03.2019

1. Ohjelman rakennesuunnitelma

Minulla tulee olemaan yksi abstrakti pohjaluokka "Object", joka käsittää kaikki pelin kannalta olennaiset asiat, kuten viholliset, ammukset ja tornit. Tulen todennäköisesti tallentamaan kaikki objektit yhteen listaan, jota päivittelen joka framella, jotta viholliset ja ammukset liikkuvat, sekä tornit kääntyvät ja ampuvat. Kaikki object-luokan perivät luokat tulevat tarvitsemaan esimerkiksi "update()" -metodin, joten se tulee olla määritettynä "object"-luokassa. Tämä tulee päivittämään niiden sijainnin, tarkistamaan onko objekti enää olemassa, ja päivittämään animaatiot, jos päätän tehdä niitä.

Viholliset, ammukset ja tornit tulevat olemaan omat luokkansa, jotka perivät luokan Object. Tornit luovat ammuksia kun viholliset ovat tarpeeksi lähellä niitä, ja niiden tehtävä on törmätä viholliseen.

Vihollisten kannalta olennaisia metodeja tulee olemaan mm. "get_hp()", "set_hp()", "get_location()" ja "move()". Jokaisen kierroksen alussa katsotaan, onko vihollisella yli 0 HP:ta, jos ei ole, se poistetaan pelistä. Muussa tapauksessa sen annetaan liikkua eteenpäin. Tornilla tulee olemaan jonkinlainen metodi joka tarkistaa, onko sen lähettyvillä vihollisia. Jos on, se muuttaa tornin boolean muuttujan "canFire" trueksi. Tämä mahdollistaa tornin alkavan ampuu while(canFire) -loopin avulla. Torneilla tulee olla muuttujina ainakin attackSpeed, damage ja range, joiden merkitys on itseselitteinen.

En ole vielä aivan varma, miten toteutan tornien ostamisen ja rakentamisen, tai vihollisaaltojen kontrolloimisen.

2. Käyttötapauskuvaus

Esitä ainakin yksi realistinen ohjelman käyttötapaus, eli kuvaus tilanteesta, jossa käyttäjä käynnistää ohjelman ja tekee sillä joitakin ohjelmalle tyypillisiä asioita. Kerro sekä se, mitä toimenpiteitä käyttäjän on suoritettava päämääriensä saavuttamiseksi (eli minkä käyttöliittymän osien kanssa tämä on tekemisissä ja miten), että se, mitkä ohjelman osat aktivoituvat kussakin vaiheessa "kulissien takana" ja suorittavat tarvittavat tehtävät. Koodiyskityiskohdat eivät ole tässä kiinnostavia, vaan korkeamman tason työnjako.

Kun käyttäjä avaa sovelluksen, se avaa jonkinlaisen menun, jossa on tavalliset menun painikkeet, kuten "start game", "load game", "quit", ja "how to play" -osio. Painamalla "start game" pelaaja pääsee peliin. Tämä aktivoi pelin pääsilman, joka jatkuu niin

kauan kunnes pelaaja joko lopettaa pelaamisen tai häviää pelin.

Pelin aikana pelaaja voi asettaa torneja kentälle raahaamalla ne ikkunan laidalla olevista kuvakkeista kentälle. Kun torni on valittuna, sen ympärillä näkyy haalea ympyrä, joka näyttää kuinka kauas torni voi ampua. Pelin alussa pelaajalla on juuri sen verran rahaa, että hän saa tehtyä yhden tornin. Tornia ostaessa ohjelma tarkistaa, että pelaajalla on tarpeeksi rahaa sen rakentamiseen, ja ilmoittaa siitä pelaajalle jos ei ole.

Painamalla "Next wave" pelaaja käynnistää ensimmäisen vihollisaallon. Tämä tarkoittaa, että x määrä vihollisia lähtee kulkemaan rataa pitkin kohdettaan kohti, ja torni pyrkii ampumaan ne ennen sitä. Tästä eteenpäin jokainen kierros tulee sisältämään enemmän ja/tai vahvempia vihollisia, kunnes lopulta puolustus pettää. Jokaisesta vihollisen kuolemasta saa rahaa, ja torneja voi ostella ja parannella kesken kierrosten ja niiden välissä.

3. Algoritmit

Tornit joutuvat ennakoimaan vihollisen liikettä ja ampumaan vihollisen nopeudesta riippuen hieman sen etupuolelle, jotta ammus osuu kohteeseen. En ole vielä varma miten teen tämän, mutta yksi vaihtoehto olisi laskea, miten pitkälle vihollinen kerkeäisi radalla siinä ajassa, jonka ammus kulkee, ja ampua ammus siihen suuntaan. Jos tämä osoittautuu hankalaksi, teen ammuksista vain sen verran nopeita, ettei ongelmia synny.

Viholliset tulevat kulkemaan tiettyä rataa pitkin siten, että siinä on vain 90 asteen kääntymisiä. Tällaisissa käännteissä on checkpoint, jonka kohdattuaan vihollinen alkaa kulkea seuraavaa checkpointtia kohti. En siis tule implementoimaan sen hienompaa matematiikkaa vihollisten kulkemiseen.

4. Tietorakenteet

Tulen todennäköisesti käyttämään vihollisten, ammusten, tornien yms. objektien tallentamiseen pelkkiä listoja. En usko tarvitsevani mitään monimutkaisempia tietorakenteita, sillä tulen päivittämään jokaisen kentällä näkyvän objektin jokaisella framella, eli käyn vain koko listan läpi.

5. Aikataulu

1. Alkusaatäminen, kuten ikkunaan piirtelyn opettelu ja siihen tutustuminen: 5h
2. Yksinkertaiset grafiikat: 2h
3. Vihollisten luominen ja niiden liike rataa pitkin: 5h
4. Tornien luominen ja niiden ampuminen: 10-20h Riippuen kuinka hankalaksi tehtävä osoittautuu. Tämä on kuitenkin pelin päämekaniikka, joten tahdon tehdä siitä mahdollisimman hyvin rullaavan.
5. Erilaisten tornien kehittäminen: 3h
6. Erilaisten vihollisten kehittäminen: 3h
7. Pelin kokonaisuuden toteuttaminen, kuten kierrokset, elämien menettäminen yms: 5h
8. Viimeistely: Tähän tulee todennäköisesti kulumaan liikaa aikaa.

6. Yksikkötestaussuunnitelma

Kuvaa tässä osiossa kuinka aiotte testata ohjelman keskeisimpiä osia toteutuksen edetessä. Koko ohjelman kaikkia ominaisuuksia ei ole tarkoitus käydä läpi, vaan keskittyä tässä ehkä ohjelman ”ydinmetodeihin”, jotka tekevät sen keskeisimmän työn. Kuvaa jälleen yleisluontoisesti, kuinka aiotte metodeja (valitkaa muutama) niiden valmistuttua kokeilla. Jälleen voi esittää keskeisiä syötteitä joilla ohjelman tulee toimia, mitä metodin tulee tällöin palauttaa ja mikä sen vaikutus ohjelman uloihin tulee olla. Vastaavasti voi pohtia sitä, kuinka metodeja voisi testata helposti ilman että tarvitsee toteuttaa valtavia apurakennelmia. (valitettavasti näiltä ei voi aina välttyä)

Yksikkötestausta kannattaa sitten projektia toteuttaessa tehdä sopivassa määrin, jotta ei turhaan joudu etsimään bugia uusista koodiriveistä, kun virhe onkin jossain joka tehtiin aikaa sitten.

7. Kirjallisuusviitteet ja linkit

Youtube, stackoverflow, wikipedia, google, sekä A+. Uskon käyttäväni kaikkia tavallisia tiedonhakulähteitä projektin aikana.