

Tornipuolustus

Yleissuunnitelma

Veeti Kuivalainen

717005

EST: 2. vuosi

Ohjelmoinnin peruskurssi Y2

6.3.2020

Yleiskuvaus

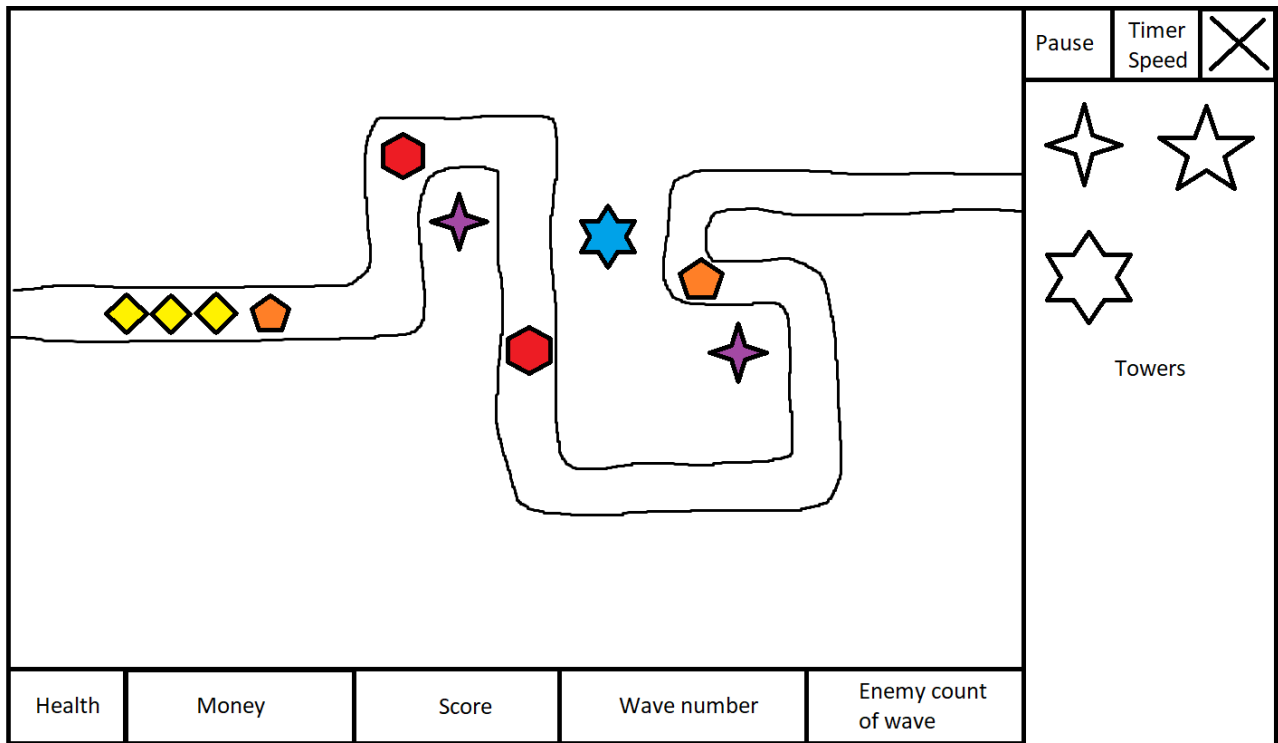
Tornipuolustuspelin tavoitteena on estää vihollisten pääseminen maaliin asettamalla polun varrelle torneja, jotka ampuvat vihollisia. Jos vihollinen pääsee vahingoittumattomana polun loppuun, pelaaja menettää elämiä. Kun elämät ovat loppu, peli on hävitty. Viholliset tulevat aalloittain ja jokaisen aallon välissä on aikaa rakentaa torneja. Jos teknisesti toteutettavissa, torneja voi rakentaa ja tuhota myös aallon aikana. Jos pelaaja selviää kaikista aalloista, hän on voittanut pelin.

Toteutan alkuun torneja, jotka ampuvat viholliset ammuksilla, jotka osuvat aina. Jos aikaa jää, yritän tehdä myös aluehyökkäyksiä, jotka osuvat jos vihollinen on hyökkäysalueella. Viholliset ovat alkuun ennalta määrättyä polkua seuraavia, jotka voivat myös jonkin tornin vaikutuksesta liikkua taaksepäin. Laajentaessa vihollisista voi tehdä esimerkiksi lentäviä tai jotain muuta polkua seuraavia. Uusien kenttien luomisen uskon olevan suhteellisen helppo homma, jonka voisin toteuttaa.

Grafiikat toteutan aluksi simppeleinä PyQt-monikulmioina, mutta olisi hienoa jos esimerkiksi kuvatiedostoja saisi ladattua peliin. Sekin olkoon yksi laajennusmahdollisuus.

Lähden tekemään projektia heti tapaamisen jälkeen (10.3). Uskon, että tornipuolustuspeli on helposti laajennettavissa, joten teen ensin pelin rungon kuntoon mahdollisimman nopeasti ja sitten jos ja kun aikaa jää niin laajennan keskivaikean-vaativan vaikeustason alueelle.

Käyttöliittymän luonnos



Kuva kertoo lähes kaiken oleellisen. Lisäksi suunnitelmissa toteuttaa pieni ikkuna, jossa näkyy valitun tai kursorin alla olevan tornin tai vihollisen tiedot. Ikkuna voi olla joko hiirtä seuraava tai oikeassa alanurkassa.

Tiedostot ja tiedostoformaatit

Erillisiin tiedostoihin voi tallettaa eri kenttiä, joissa käytännössä on kentän ulottuvuudet ja vihollisen seuraama polku. Näissä tiedon esittämiseen tarvitsee vain kaksi saraketta, joten csv-tiedosto olisi näppärä. Lisäksi teen tiedoston, jossa on vihollisaaltojen ominaisuudet, kuten vihollisen määrä, tyyppi, elämät ja sen sellaiset. Tämän voisi tehdä json-tiedostona. Nämä tiedostot voi ehkä yhdistää, mutta alkuun pidän erillisinä, koska teen vihollisaaltotiedoston ensin ja uusien kenttien lisäyksen myöhemmin.

Järjestelmätestaussuunnitelma

Pelille annettavien kenttätiedostojen täytyy olla ehjiä. Siksi pitää olla testit, jotka estävät esimerkiksi polun, joka risteää itsensä kanssa tai päättyy ruudun ulkopuolelle. Vihollisaaltojen ominaisuustiedoston pitää myös olla luettavissa ja järkevä, esimerkiksi vihollisten määrä ja elämät eivät voi olla negatiivisia. Täytyy siis testata pelin toimintaa erilaisilla rikkiäisillä tiedostoilla.

Itse pelin toiminnallisuuteen liittyen testaus tulee pitkälti peliä ajamalla ja havainnoimalla. Kuitenkin joitain asioita, kuten päällekkäisiä torneja tai talouteen liittyviä asioita täytyy testata pythonin testityökaluilta.