

**Tornipuolustus**  
**Harjoitustyö**  
**Ohjelmoinnin peruskurssi Y2**  
**CSE-A1121**

Paavo Pärssinen

356644

AIT 2014

5.5.2016

## Yleiskuvaus

Tavoitteena oli tehdä yksinkertainen tower defence – tyylinen peli. Pelissä joukko vihollisia pyrkii pääsemään maaliin ennalta määrättyä reittiä pitkin. Pelaajan tavoitteena on estää vihollisen maaliin pääseminen asettamalla reitin varrelle erinäisiä torneja, jotka ampuvat tai jollain muulla tavalla haittaavat hyökkääjiä.

Peliin tarvitaan luonnollisesti graafinen käyttöliittymä, mutta muuten grafiikat saavat olla melko alkeellisia. Peliä pystyy myös helposti laajentaa asetustiedostojen kautta. Grafiikoiden luomiseen on käytetty Pythonin pygame-moduulia.

Tavoitteena olisi saada peliin erikokoisia vihollis-aaltoja, jotka vaikeutuvat ajan kuluessa. Pelaaja saa aina voittaessaan rahaa, jolla voi ostaa uusia torneja, sekä pisteitä joilla voidaan vertailla eri pelaajien suoritusta.

## Käyttöohje

Ohjelma ajetaan ajamalla main-tiedosto. Peli lähtee suoraan käyntiin, jolloin ensimmäinen aalto vihollisia alkaa liikkua lähtöpisteestään. Pelaajalla on pieni määrä rahaa, jolla hän voi rakentaa ensimmäiset torninsa. Peliruudun alareunassa on harmaa ikkuna, joka sisältää useita erilaisia toimintoja. Alavasemmalla oleva ikkuna antaa palautetta pelaajan toimista; esimerkiksi miksi tornia ei voi rakentaa tähän, tai päivitys rakennuksen onnistumisesta. Palauteikkunan oikealla puolella on kaksi painiketta; ”Build” ja ”Cancel”. Painamalla ”Build”-painiketta pelaaja pääsee rakennustilaan, ja painamalla ”Cancel” pelaaja pääsee pois tilasta.

Rakennustilassa ollessaan pelaajan alapalkkiin ilmestyy neljän eri tornin painikkeet. Painikkeissa lukee tornin nimi, sekä näkyy tornin ulkonäkö. Painamalla tornia päästään kyseisen tornin rakennustilaan. Kyseisessä tilassa muiden painikkeiden oikealla puolelle ilmestyy laajennettu tietopaketti valitusta tornista. Tällöin näkyville tulee suurempi kuva tornista, sekä sen erilaiset ominaisuudet kuten ampumaetäisyys ja nopeus. Siirtäessään cursorinsa pelikentälle pelaaja näkee nyt mihin on rakentamassa sitä, sekä myös sen ampumaetäisyyden harmaana ympyränä. Painamalla hiiren vasenta näppäintä pelaaja voi rakentaa tornin. Pelaaja pääsee pois kyseisen tornin rakennustilasta klikkaamalla itsensä joko toisen tornin rakennustilaan, tai painamalla ”Cancel”.

Alapalkin oikeassa kulmassa on lisäksi tiedot pelaajan nykyisestä rahatilanteesta, kuluneesta ajasta, pistetilanteesta, tämänhetkisen vihollis-aallon numero sekä rakennettujen tornien määrä. Tornien määrä on tällä hetkellä rajoitettuna kymmeneen; vanhojen tornien purkaminen tapahtuu painamalla niitä hiiren oikealla painikkeella, jolloin pelaaja saa puolet niihin sijoittamistaan rahoista takaisin.

Peli päättyy kun vihollinen pääsee maaliin. Tällöin pelilaudan päälle ilmestyy tekstinsyöttöä varten ikkuna, johon pelaaja voi syöttää nimensä. Painamalla Enter pelaaja pääsee katsomaan Top 10 – pelaajien listaa, sekä mahdollista sijoitustaan siellä. Painamalla mitä tahansa näppäimistön painiketta koko ohjelma sulkeutuu.

## Ohjelman rakenne

Peli sisältää useita luokkia pääluokan lisäksi. Game-luokka sisältää pelilaudan joka itsessään on Board-luokka. Tämä luokka taas sisältää laudallisen Spot-olioita; joiden alaluokat Tower ja Road kuvastavat torneja ja vihollisen reittiä. Lisäksi erikseen on Enemy-luokka, joka kuvastaa erilaisia vihollisia. Alla on lyhyt kuvaus luokista.

Luokat:

- main
  - Luokka mallintaa koko peliä, sisältää tavallaan kaikki muut peliin sisältyvät luokat.
  - Metodeihin kuuluu esimerkiksi pelin aloitus, tallentaminen, sekä kaikki klikkauksiin perustuvat pelin toiminnot (rakentaminen jne.)
  - Laskee kierroksia, jotka määräävät vaikeustason
  - Sisältää pelaajan tiedot: rahat, pisteet, taso jne.
  - Metodeja:
    - tornien, vihollisten, kartan sekä vihollisaaltojen lataaminen tiedostosta
    - ruudun päivityksen metodi, joka jaettu useisiin eri osiin
- Game
  - Sisältää Board-luokan
  - Suuri osa suunnitellusta sisällöstä löytyy tällä hetkellä main-luokasta
- Board
  - Sisältää 20\*12 taulukon Spot-olioita
  - Main-luokan metodit rakentamiseen ja purkamiseen käyttävät tämän luokan vast. metodeja
  - Laskee laudalla olevia torneja
- Spot
  - Alaluokka Tower
    - Tietää kaikki ominaisuutensa (nopeus, voima, hinta jne.)
    - Tietää tähtäimessään olevat viholliset
    - Metodeja mm. kulman laskemiseksi viholliseen
    - Metodeja joita käytetään ampuma-animaation luomiseen
    - Metodi ampumiseen
  - Alaluokka Road
    - Tietää paikkansa laudalla, sekä seuraavan Road-olion
    - Tietää onko alku- tai loppupiste reitillä
- Enemy
  - Ominaisuuksina mm. HP, nopeus, ja palkintosumma
  - Metodeina esim. liikkuminen seuraavaan ruutuun laudalla (riippuu tietenkin yksilön nopeudesta), sekä osuman ottaminen (HP vähenee).
  - Metodeina on lisäksi oman animaation luomiseen tarkoitettuja metodeita

## Algoritmit

Vihollinen tietää aina missä ruudussa on ja mikä on seuraava, vihollisen nopeus määrää ajan joka kestää siirtyä toiseen ruutuun. Pelin toiminnassa taustalla on käytetty lähes kaikkialla `pygame.time.get_ticks()`-metodia, joka kertoo ajan. Myös animaation piirtäminen on suoritettu siten, että ruutua päivitetessä viholliselle kerrotaan tämänhetkinen aika, jonka perusteella vihollinen piirretään oikeaan pisteeseen.

Tornin animaation kannalta tärkein algoritmi on kulman laskeminen tähtäimessä olevaan viholliseen. Tämä lasketaan käyttämällä vihollisen ja tornin sijaintia sekä `math.atan2()` funktiota. Lisäksi tornien laskiessa mahdollisia vihollisia tähtäimeen lasketaan Pythagoraan lauseen mukainen etäisyyden laskeminen kahden pisteen välillä. Pisteillä siis  $x$ - ja  $y$  – koordinaatit, jolloin kahden pisteen etäisyys saadaan näin:

$$\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}.$$

## Tietorakenteet

Suuri osa ohjelmasta käyttää Pythonin dictionary-rakennetta. Totesin tämän olevan array-rakennetta helpompi, sillä siihen on helpompi lisätä alkioita ja poistaa niitä (on siis muuttuvatilainen). Käytin niitä usein muuten samalla tavalla kuin array-rakennetta, eli luomalla alkioden avaimiksi esimerkiksi luvut yhdestä kymmeneen.

## Tiedostot

Ohjelma käsittelee txt-tiedostoja, jotka ovat suunnitellun muotoisia. Ohjelmassa on erikseen metodit neljän eri tower-tyypin lukemiseen tiedostosta, vihollistyyppien lukemiseen tiedostosta, sekä kartan ja vihollisaaltojen tietojen lukemiseen. Lisäksi ohjelmalla on metodi top 10 – pelaajien lukemiseen ja kirjaamiseen.

Kaikki luettavat tiedostot on kirjoitettu niin, että rivi riviltä luetaan eri osa tiedoista, jotka sitten lisätään ohjelman dictionary-rakenteisiin. Suurin osa tekstitiedoista on myös ymmärrettävissä vain lukemalla tiedostoja.

## Testaus

Panostin testauksessa tärkeintä eli vuorovaikutusta tornin ja vihollisen välillä; tämä kuului myös alustavaan suunnitelmaani. Ensimmäisessä yksikkötestissä testataan etäisyyden laskemista viholliseen, toisessa vihollisen löytämistä kun se on tarpeeksi lähellä ja kolmannessa vihollisen löytämistä kun se on liian kaukana. Lisäksi otin mukaan vihollisen ampumisen, sekä sen kuoleamisen tarkastamisen.

Ohjelma läpäisee nämä testit. Olen itse käyttänyt melko vähän yksikkötestauksen keinoja ohjelmaa rakennettaessa, itse olen tottunut esimerkiksi laittamaan ohjelman printtaamaan tietoja asioista jotka kiinnostavat minua, ja täten korjannut virheet. Nyt kun tätä testausta varten tein näitä yksikkötestejä, huomasin kuitenkin että monet asiat olisi ollut helpompi tutkia näillä. Tulen vastaisuudessa kokeilemaan ensiksi yksikkötesteillä.

## Tunnetut puutteet ja viat

Listattuja puutteita ja vikoja:

- main-luokka ei ole kovin kauniisti järjestetty, hieman liian suuri osa ohjelman toiminnasta on metodien ulkopuolella
- Game-luokka jäi pitkälti käyttämättä, suuri osa suunnitelluista toiminnoista onkin main-luokassa
- Animaatiota ei ole sidottu mihinkään fps-nopeuteen. Aiheuttaa hieman turhaa suoritustehon käyttöä
- Bugi: pelin edetessä noin aaltoon 5, alkaa näytöllä olla todella paljon vihollisia. Tällöin jostain syystä jotkin viholliset saattavat alkaa kuolla yhdestä ammuksesta
- Tornien ammukset laskevat suuntansa uudestaan joka hetki, syntyy pieni tarkoitukseton "hakeutuva ammus"-effekti
- Kaikkia virheellisiä luettavia tiedostoja ei osata käsitellä kaatumatta

## Kolme parasta ja heikointa kohtaa

Parhaat:

- Tornien kääntyminen vihollista kohti toimii nätisti
- Eri pelaajien tuloksia helppo verrata Hall of Fame –osiossa pelin jälkeen
- Tornien rakentaminen: mm. infoikkuna valitusta tornista, sekä haamukuva ampumaetäisyydestä

Heikoimmat:

- Kun ruudulla useita vihollisia (n. 100), saattaa peli hidastella sekä bugeja ilmaantua
- Vihollisten tarkka sijainti olisi ollut fiksumpaa laskea metodilla vihollisten sisällä, nyt ympäri ohjelmaa on turhaan sama hankala pätkä koodia
- Ohjelman yleinen epäjärjestys: useat ratkaisut ei ole suunniteltu miettien tulevia vastaavia ongelmia, jolloin jouduttu luomaan lisää ja lisää metodeja tuleviin ongelmiin

## Poikkeamat suunnitelmasta

Suurin muutos suunnitelmasta oli kenties pyQT:n vaihtaminen pygame-moduuliin.

Harjoitustyön tekemisen aikana luovuin muutamista suunnitelluista ominaisuuksista. Esimerkiksi tornien siirtäminen ja päivittäminen korkeammille tasoille on jätetty pois. Myös pelaajan kokemuspisteiden on kerääminen jätetty pois; olisi kuitenkin helppo lisätä sillä pelaaja kerää kyllä pisteitä, ne eivät vaan tuo uusia leveleitä.

Lisäksi jätin pois pelin alkuvalikon, jossa olisi voitu esimerkiksi valita kenttä tai muuttaa asetuksia. Nämäkin olisi melko helppo vielä lisätä, mutta jää tekemättä ajanpuutteen takia. Suunnitelmassa Game-luokalla olisi ollut isompi rooli: nyt suurin osa koodista main-luokassa.

## Toteutunut työjärjestys ja aikataulu

Arvioin alussa että projektia tulee tehdä noin 8 tuntia viikossa; tämä on jokseenkin pitänyt paikkansa. Kuitenkin muutama viikko jäi välistä projektin suhteen, jolloin seuraavina viikkoina on saanut ohjelmoida yötä myöten.

Alkuviikot kuluivat suunnitelman mukaan vain graafisen käyttöliittymän käytön harjoitteluun, jonka jälkeen loin raakaversiot kaikista luokista. Animaatioiden luominen taas syntyi jo noin viikon 5. kohdalla, toisin kuin olin ajatellut. Viimeiselle viikolle jäi hieman liikaa tekemistä: mm. koko dokumentti, pelin vaikeustason optimointi sekä yksikkötestit. Aikataulun tiukkuuden vuoksi viimeistelyä ja bugien siivoilua saattaa jäädä tekemättä.

## Arvio lopputuloksesta

Olen itse erittäin tyytyväinen tulokseen, ottaen huomioon kyseessä olevan ensimmäinen graafinen ohjelmani. On ollut hauskaa oppia erilaisia kikkoja animaatioiden aikaansaamiseksi, sekä oppia ymmärtämään pelien peruseriaatteita.

Työssä ei mielestäni ole mitään kriittisiä puuttuvia osia: pelissä on toimiva graafinen käyttöliittymä, mukavasti toimivat grafiikat sekä ohjelman toiminnan helppo laajennettavuus txt-tiedostoilla. Lisäksi olen melko tyytyväinen wave-systeemiin, joka syöttää vihollisia tiettyyn tahtiin eri kierroksilla.

Ohjelmaan on helppo keksiä paljon lisää lisättävää. Esimerkiksi juuri alkuvalikko olisi helppo lisätä pyörimään ennen pääikkunan käynnistymistä. Olisi helppoa myös lisätä vaihtoehtoisia grafiikoita ja kenttiä valikkoon. Tärkeää pelin pelattavuuden kannalta olisi myös uusien tornien ja vihollisten lisääminen; paremmassa pelissä olisi esimerkiksi "area damagea" –aiheuttava torni, tai vihollisia hidastava piikkimatto, lentäviä vihollisia jne. Osa näistä olisi melko helppo lisätä mutta monet vaatisivat useamman tunnin kovan koodauksen. Myös exp-systeemi ja levelit pelaajalle sekä torneille olisivat hyviä lisäyksiä.

Luokkajaossa voitaisiin siirtää suuri osa koodista pois main-luokasta. Esimerkiksi Game-luokka olisi paljon parempi paikka useille metodeille. Lisäksi joissain kohdissa olisi ollut paljon fiksumpaa luoda metodi kuin kirjoittaa sama pitkä pätkä koodia uudestaan.

Ohjelman rakenne mahdollistaa helpot muutokset; tekstitiedostoja muokkaamalla peliä saadaan helposti tasapainotettua ja vaihdettua erilaisia torneja muiden tilalle. Laajentaminen kuitenkin vaatisi paljon enemmän, sillä peli tukee vain neljää tornia ja pitkälti myös vain neljää eri vihollista. Vihollisten lisääminen olisi melko helppoa muuttamalla koodia hiukan, mutta torneilla vaadittaisiin rakennusvalikon suurta laajentamista.

## Viitteet

- Yleisesti Tower defence – pelaista
  - <http://www.raywenderlich.com/37701/how-to-make-a-tower-defense-game-tutorial>
- Yleisesti Pythonista
  - [http://www.tutorialspoint.com/python/python\\_quick\\_guide.htm](http://www.tutorialspoint.com/python/python_quick_guide.htm)
  - <https://plus.cs.hut.fi/y2/2016/>
  - <http://www.cse.hut.fi/fi/opinnot/CSE-A1111/Kesa2014/oppimateriaali.html>
- pygame
  - <http://www.pygame.org/docs/>
  - <https://inventwithpython.com/pygame/chapters/>
- Muita
  - <http://stackoverflow.com/>
  - [http://www.tutorialspoint.com/python/python\\_dictionary.htm](http://www.tutorialspoint.com/python/python_dictionary.htm)