

20.4.2017

# **Ohjelmointistudio 2: Projekti**

## **Tornipuolustuspeli**

Alexi Siitari  
594503  
Tietotekniikka 2016

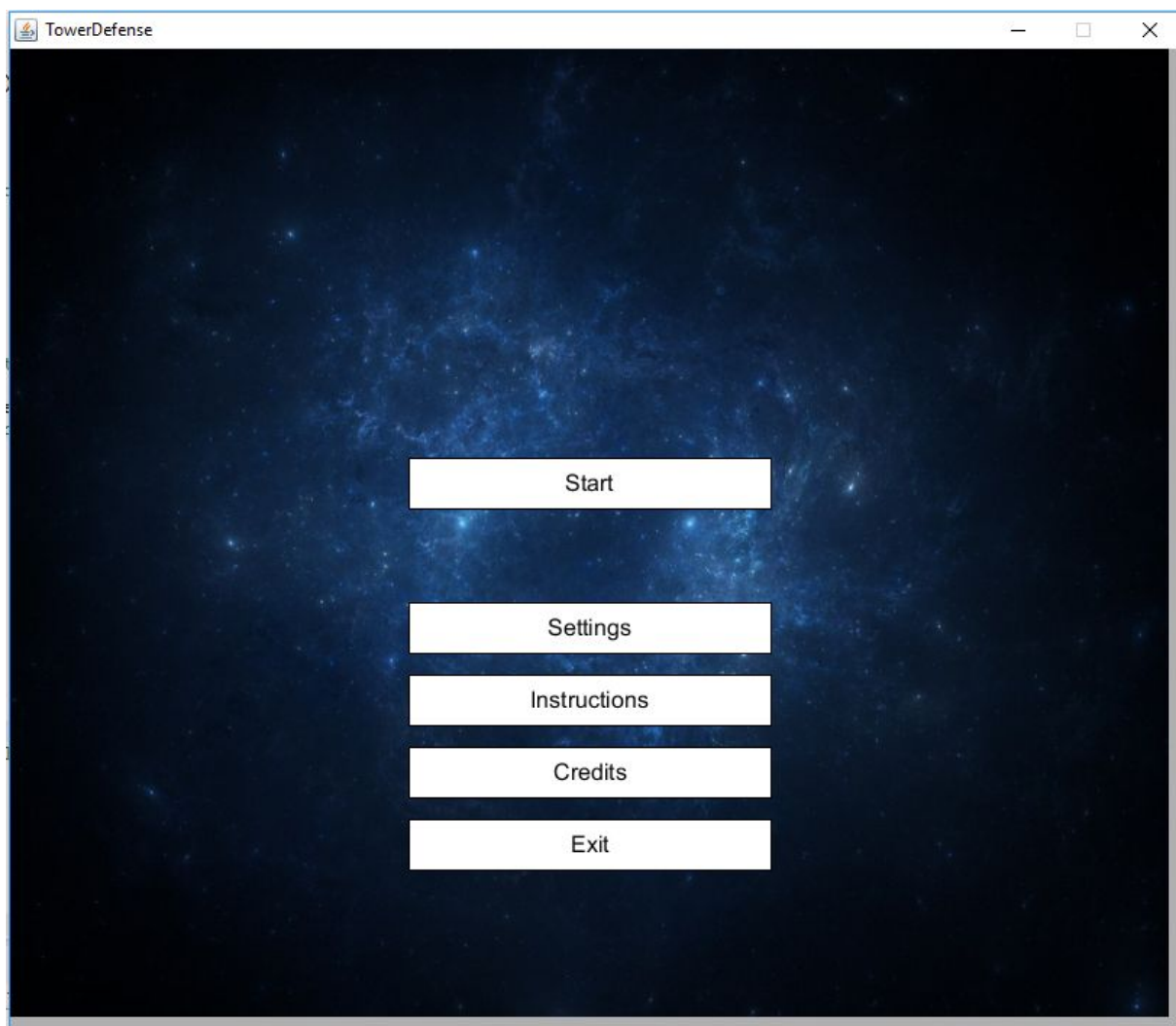


## 1. Yleiskuvaus

Projekti on reaaliaikainen tornipuolustus -peli, jossa pelaajan on torneja rakentamalla ja taikoja käyttämällä tarkoitus estää vihollisia kulkemasta tason läpi. Viholliset hyökkäävät aaltoina, jotka vaikeutuvat pikkuhiljaa. Aaltojen välissä pelaajalla on hetki aikaa hengähtää. Pelaajan tuhotessa vihollisia hän saa käyttöönsä lisää rahaa, jolla voi rakentaa torneja tai käyttää taikoja. Tarpeeksi monen vihollisen päästyä kentän läpi, pelaaja häviää.

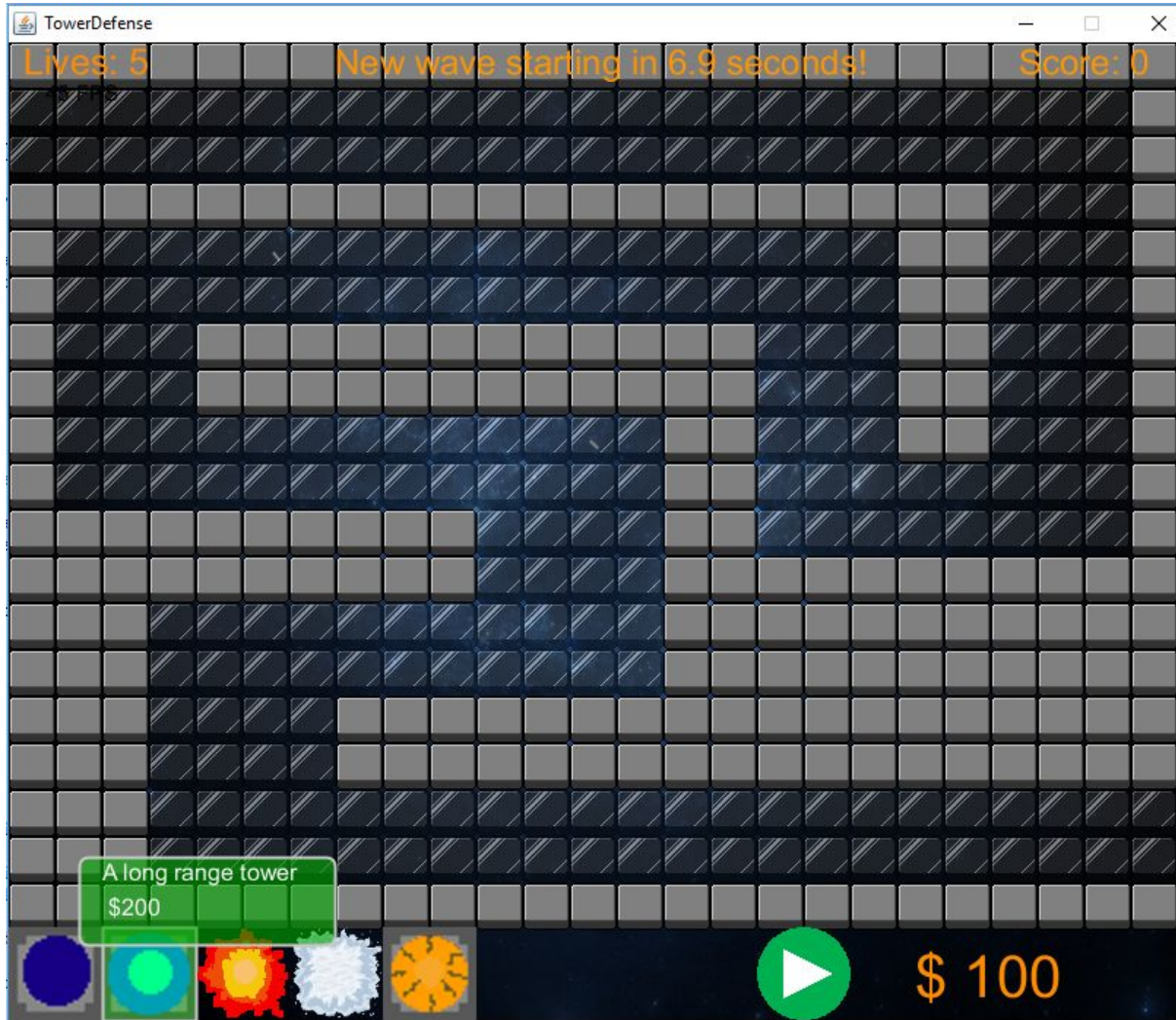
## 2. Käyttöohje

Ohjelma Käynnistetään TD.scala tiedoston sisällä olevasta main oliosta. Peli käynnistyy päävalikkoon josta on mahdollista navigoida itse peliin, ohjeisiin, asetuksiin tai creditseihin.





Pelissä pelaaja voi hiiren vasemmalla painikkeella valita haluamansa tornin tai taian ruudun alaosassa olevasta palkista ja samalla painikkeella rakentaa kyseisiä torneja kentälle. Oikea hiiren painike poistaa tornin tai taian valinnan. Lisää tietoa torneista ja taioista saa pitämällä hiirtä kuvakkeen päällä muutaman sekunnin, jolloin ruudulle ilmestyy tooltip kyseisestä painikkeesta.



Pelin alalaidassa näkyy myös pelaajan varat, joita käytetään tornien rakentamiseen ja taikoihin sekä vihreä play -nappula, josta voi aloittaa seuraavan vihollisaallon ilman odotusta. Ylälaidassa näkyy pelaajan elämät, eli kuinka monta vihollista voi kulkea tason läpi ennen kuin pelaaja häviää, aika seuraavaan vihollisaaltoon ja pelaajan pisteet.

### 3. Ohjelman rakenne

Ohjelma jakautuu viiteen pakettiin, jotka sisältävät hieman erilaisia luokkia ja olioita. Modes pakettiin sisältyy ohjelman eri tilojen toteutus. Esimerkiksi päävalikko avautuu kun muuttuja mode vaihdetaan main oliossa MenuModeksi ja itse peli alkaa kun se muutetaan PlayModeksi. PlayMode on olio, joka pääasissa pitää kirjaa kaikesta itse pelin sisällä tapahtuvasta kuten vihollisista, torneista, kartasta jne. Jokaisella modella



on init ja terminate funktiot, joilla voidaan ongelmitta vaihtaa tilaa. Draw ja Update funktiot taas huolehtivat että kyseinen tila piirretään ja päivitetään.

Main olio pitää kirjaa mitä Modea näytetään ja sisältää itse game loopin draw funktiossa, jonka sisällä koko peli pyörii. Main oliossa määritetään myös framerate eli kuinka monta kertaa peli piirtää uuden kuvan ruudulle ja päivittää pelin tilan.

## 4. Käytetyt algoritmit

Pelissä ei ole kovinkaan monimutkaisia algoritmeja käytössä. Polunseuranta hoituu yksinkertaisella pisteestä toiseen kulkemisella. Kun vihollinen on päässyt tarpeeksi lähelle sen kohde pistettä, se siirtyy liikkumaan kohti seuraavaa pistettä. Lopulta kun vihollinen saavuttaa viimeisen pisteen se on saavuttanut lopun, jolloin pelaaja menettää elämän ja vihollinen tuhoutuu. Kuljettavat pisteet ladataan tiedostosta path.txt pelin alussa, joten ne ovat konfiguroitavissa erilaisiin tasoihin.

Osuman tarkistus tapahtuu vertailemalla jokaisen ammuksen sijaintia jokaisen vihollisen sijaintiin. Algoritmi on hieman työläs, mutta tehokkaamman keksiminen on vaikeaa. Eräs mahdollinen ratkaisu olisi verrata ammuksen sijaintia vain niiden vihollisten sijaintiin jotka ovat samalla alueella kartalla. Kuitenkin ammusten lukumäärä pelissä on suuruusluokaltaan niin pieni ettei monimutkaisempaa toteutusta tarvita.

## 5. Tietorakenteet

Lähes kaikki ohjelman ajonaikaisista tiedoista säilytetään Buffereissa, joihin on helppo lisätä ja poistaa tietoa ajonaikana. Osaa tiedoista olisi voinut säilyttää yksinkertaisemmissakin tietorakenteissa, mutta siitä ei olisi saatu merkittävää hyötyä.

## 6. Tiedostot

Suurinosa pelin tarvitsemista tiedostoista on kuvia jotka on tallennettu resources kansion sisällä olevaan images kansioon. Kuvat ovat .png muodossa, jotta voidaan käyttää alpha läpinäkyvyyttä. Kuvien koko ei nouse liian suurensi sillä pelin resoluutio on 800x670 ja tornien ja vihollisten kuvat ovat suuruusluokkaa 32x32.

Pelin kartta ja vihollisten käyttämä reitti [luetaan myös tekstitiedostoista](#) eli ne ovat muokattavissa.

## 7. Testaus

Projektin testaus tapahtui pääasiassa ajamalla ohjelmaa, sillä lähes kaikki mahdolliset virheet oli huomattavissa ajon aikana helposti.





Vector luokkaa testattiin erillisillä UnitTesteillä. Suurinosa pelin laskuista käyttää Vector luokkaa, mahdolliset bugit on vaikea havaita ja niillä on suuri vaikutus pelin toimintaan. VectorTestistä löytyvät UnitTestit käyvät läpi Vector luokan tärkeimmät toiminnot ja varmistavat että ne toimivat oikein.

## 8. Puutteet ja tunnetut viat

Pelissä suurimmat puutteet liittyvät pelin optimointiin, Cursor olioon ja ominaisuuksien määrään. Näitä kolmea tekijää olisi helppo parantaa, jos aikaa olisi enemmän.

Peli lakkaa toimimasta sulavasti kun ruudulle tulee tarpeeksi monta vihollista ja tornia. Erityisesti peli hidastuu, jos pelaajan kursori on tornin päällä jolloin kaikkien tornien kantamat piirretään näytölle. Suorituskykyä voitaisiin parantaa esimerkiksi piirtämällä vain sen tornin kantama, jonka päällä pelaajan kursori on. Myös pelin kartan voisi luoda yhdeksi kuvaksi pelin alussa jolloin Processingin Draw funktiota tarvitsisi kutsua vain kerran koko kartan piirtämiseksi. On tehokkaampaa kutsua yhtä Draw funktiota isolle kuvalle kuin monta funktiota pienelle kuvalle.

Cursor olion toteutus olisi voinut olla yksinkertaisempi, jolloin peliin olisi nopeampi lisätä uusia ominaisuuksia ja siinä olisi vähemmän toisteista koodia.

Pelin ominaisuudet jäivät myös mielestäni vähäisiksi. Peli olisi kaivannut enemmän vaihtelevia torni- ja vihollistyyppejä, taikoja ja tasoja. Näitä uusia ominaisuuksia olisi ollut helppo lisätä peliin enemmän.

Pelissä on myös muutama pieni bugi kuten vihollisaallon loputtua tuleva teksti "Wave x has ended", joka muuttuu jos pelaaja aloittaa seuraavan vihollisaallon vihreästä play painikkeesta. Esimerkiksi kun pelaaja on voittanut aallon numero 2, ruudulla lukee "Wave 2 has ended". Nyt jos pelaaja painaa vihreästä play napista kun teksti vielä on näkyvillä, ruudulle tulee lukemaan "Wave 3 has ended". Bugi johtuu Spawner -olion laiskasta toteutuksesta ja se olisi hyvinkin mahdollista korjata esimerkiksi olemalla päivittämättä tekstiä silloin kun se näytetään ruudulla.

Myös Ice abilityn toteutus jäi mielestäni todella huonoksi. Sen tarkoitus olisi hidastaa vihollisia vain sen ajan kun ne ovat jään vaikutusalueella, mutta tällä hetkellä jää vain muuttaa vihollisen nopeuden pysyvästi tiettyyn arvoon. Tämän voisi toteuttaa paremmin esimerkiksi tekemällä uuden effect -luokan, joka vaikuttaisi aina tiettyyn Enemy olioon tietyn aikaa. Tällaista toteutusta voisi käyttää myös Fire abilityyn ja se parantaisi ohjelman laajennettavuutta.

Joskus myös viholliset liikkuvat seinien päällä. Tätä tapahtuu erityisesti Mothership vihollistyyppillä ja se olisi helppo korjata esimerkiksi lisäämällä osuman tunnistuksen myös seinien ja vihollisten välille. Jos vihollinen liikkuisi tarpeeksi lähelle seinää, seinä alkaisi hylkiä vihollista ja työntäisi sen takaisin reitilleen. Toisaalta tällaisen algoritmin lisäys vaatisi paljon laskentatehoja eikä tuottaisi merkittävää hyötyä.



## 9. 3 parasta ja heikointa kohtaa

### + Graafinen ulkoasu ja käyttökokemus

Olen erittäin tyytyväinen pelin ulkoasuun ja peliä on mielestäni helppo käyttää. Tooltipsit ja muut graafiset indikaattorit ovat selkeitä eivätkä jätä pelaajaa ihmettelemään. Ainoastaan pelinäkymän yläosassa olevat tekstit jotka kertovat pelaajan elämät, pisteet ja vihollisaallon numeron ovat huonot, eivätkä erotu kunnolla taustasta. Jos aikaa olisi ollut enemmän, olisin laajentanut ikkunan kokoa ja lisännyt ylös erillisen palkin johon olisin sijoittanut tiedot silmää miellyttävällä tavalla.

### + Pelin vakaus

Pelissä ei juurikaan ole bugeja ja sen toiminta on hyvin vakaata. Suurimmat havaitsemani bugit liittyvät HUDin teksteihin eivätkä ole kovinkaan suuria. En ole myöskään saanut peliä kaatumaan mikä on hyvä merkki.

### + Peliä on hauska pelata

Huomasin projektin aikana ja sitä testatessa että pelin pelaaminen on hauskaa. Välillä peliä testatessani unohduin pelaamaan hieman pidemmäksi aikaa kuin oli tarkoitus.

### - Uusien karttojen tekeminen

Pelin kartan ja vihollisten polun lataajat ovat erittäin herkkiä virheille, sillä ne lukevat syötteen suoraa eivätkä huomioi mahdollisia virheitä. Tiedostojen lataajien olisi hyvä antaa pientä varaa virheille. Tiedostoista olisi voinut myös tehdä helpommin luettavia. Olisi myös kätevää että peli osaisi ladata useamman kentän ja polun kerralla, jolloin pelin sisältä voisi valikosta valita taso.

### - Vihollisten polunseuranta

Vihollisten polunseuranta on toteutettu alkeellisella pisteestä pisteeseen kulkemisella. Tämä tapa on varmatoiminen, mutta se täytyy tehdä jokaiselle tasolle erikseen ja on herkkä virheille. Monimutkaisempi polunetsintä olisi helpottanut laajennettavuutta, koska pelkkä tason tekeminen olisi riittänyt eikä vihollisten seuraamaa polkua olisi tarvinnut määrittää.

### - Konfiguroitavuus

Peliä ei nykyisessä muodossaan pysty juurikaan konfiguroimaan uusien karttojen ja vaikeusasteen säädön lisäksi. Jos aikaa olisi ollut enemmän, olisin lisännyt asetusvalikkoon mahdollisuuden konfiguroida enemmän pelin eri osa-alueita kuten esimerkiksi aloitusrahoja, pelaajan elämiä ja tornien ja vihollisten eri lukuarvoja.



## 10. Poikkeamat suunnitelmasta

Suunnitelmanani oli alunperin toteuttaa polunetsintä algoritmi, jolla vihollishahmot löytävät reitin tason läpi, mutta projektin aloitettuani totesin että yksinkertainenkin polunseuranta riittää. Polunseurannassa viholliset seuraavat ennaltamäärättyjä pisteitä ja siirtyvät pisteen saavutettuaan liikkumaan kohti seuraavaa. Tämä menetelmä on helpompi toteuttaa, mutta uusia tasoja lisätessä täytyy määrittää myös reitti (lista koordinaatteja) tason läpi.

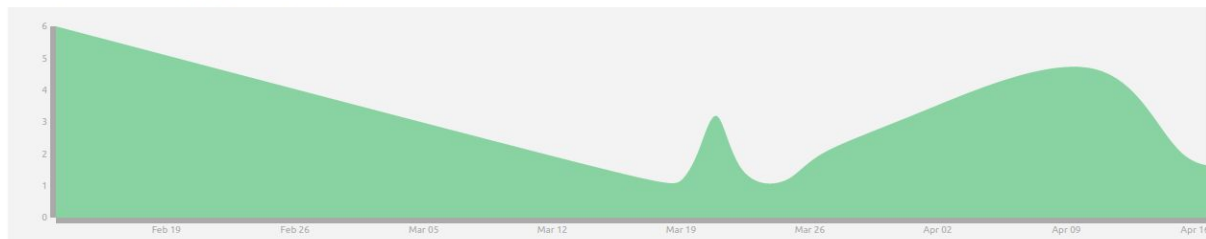
Suunnitelmasta poiketen peliin ei myöskään tullut ääniä, sillä koin että tärkeämpää oli saada hiottua jo olemassaolevat toiminnot kuntoon. Äänien lisääminen ei olisi tuonut merkittävää lisäarvoa suhteessa siihen vaadittavaan työmäärään.

## 11. Toteutunut työjärjestys ja aikataulut

Projektin toteutus ei täysin seurannut aikataulua, mutta työjärjestys oli lähes sama kuin teknisessä suunnitelmassa. Alussa toteutin pohjan kaikille luokille, joita täydensin projektin edetessä. Sen jälkeen tein yksinkertaisen polunseurannan, jolloin peliä pystyi jo testaamaan. Seuraavaksi lisäsin yksinkertaisen käyttöliittymän ja ominaisuuksia. Loppuaika kului käyttöliittymää hioessa ja ominaisuuksien ja grafiikoiden päivittämisessä.

February 13 2017 - April 21 2017

Commits to master, excluding merge commits. Limited to 6,000 commits.



Ohessa on graafi GitLabista joka näyttää committien määrän ajan funktiona. Graafi antaa suuntaa antavan arvion siitä milloin työtä on tehty. Alun boilerplate koodin kirjoittamisen jälkeen innostus hiipui ja projekti hautautui muiden koulutöiden alle.

Kuitenkin loppua kohden tekeminen kiihtyi kun muut koulutyöt helpottivat ja deadline alkoi lähestyä. Lopuksi jäljellä oli myös paljon ominaisuuksien lisäämistä ja bugien korjailua, josta kertyi paljon committeja.



## 12. Arvio lopputuloksesta

Itse olen varsin tyytyväinen lopputulokseen, vaikka lopputuloksesta tuli hieman erilainen kuin suunnitelmasta. Toisaalta on hyvä etten tarkasti noudattanut suunnitelmaa, sillä projektia tehdessä löysin parempia ratkaisuja enkä jäänyt jumiin pohtimaan hankalia ja tehottomia ratkaisuja. Koin että opin paljon peliä tehdessä ja palauttaessa jäi tunne että peliä olisi vielä vähän voinut hioa ja lisätä ominaisuuksia.

Ohjelman laajennettavuus on suhteellisen hyvä, mutta muuttaisin vihollisten reitinseurannan reitinsinnäksi jolloin laajennettavuus paranisi. Luokkajaossa olisi myös vähän parantamisen varaa, sillä se on mielestäni hiukan sekava ja monet luokat viittaavat moneen toiseen luokkaan.

## 13. Viitteet

<http://natureofcode.com/>

<https://processing.org/>

[https://plus.cs.hut.fi/studio\\_2/2017/k15/osa03/](https://plus.cs.hut.fi/studio_2/2017/k15/osa03/)

## 14. Liitteet

