Antti Tiusanen 742261 Tietotekniikan kandidaatti vsk 2021, 27.04.2022

Yleiskuvaus:

Lopullinen tulos poikkesi suunnitelmasta aika paljon aika- ja oman ymmärryksen rajoitteiden takia. Tein tornipuolustuspelin, missä on torneja, vihollisia ja kartta, joka toimii sekä kulkureittinä vihollisilla ja alustana torneille. Pelin parametrit ovat helposti muokattavissa.
Työ on simppeli käytännössä ja sen muokattavuus tekee työstä hyväntasoisen. Viimeistely kuitenkin jäi hieman vähiin, joten parannettavaa kyllä löytyy (esim. ammukset ja omat grafiikat puuttuvat).

Käyttöohje:

Ohjelma käynnistyy yksinkertaisesti GUI tiedoston ajamisella. Kun graafinen liittymä aukeaa. Voi uuden pelin aloittaa “Start Game” painikkeella tai jos vanha tiedosto on olemassa toimii myös “Load Game”. Viholliset kulkevat automaattisesti rataa pitkin ja ne eivät saa päästä radan läpi. Torneja voi ostaa mikäli kulta riittää siihen. Tornien napit vastaavat eri torneja ja klikkaamalla niistä kerran tulee esille niiden tiedot. Jos torni on valittu ja raha riittää voidaan ne asettaa vihreälle alueelle, jossa ne automaattisesti vahingoittavat vihollisia, jotka ovat kantamalla. Vihollisten määrä ajan myötä kasvaa ja vihollis aaltojen loppuessa tai “HP”:n loppuessa peli päättyy. “Testi” nappi ei tee mitään tällä hetkellä, mutta siihen voi liittää toimintoja. “>” nappi nopeuttaa peliä kaksinkertaiseen nopeuteen ja “>>” vastaavasti hidastaa sen takaisin.

Ohjelman rakenne:

Graafinen liittymä:

Ohjelman graafinen liittymä jakautuu luokkiin GUI, LevelGUI, EnemyGUI ja TowerGUI. Nimiestä mukaisesti nämä osat vastaavat omien konkreettisten luokkiensa piirtämisestä. GUI taas piirtää ikkunan ja vastaa nappien kuuntelemisesta, hiirten painelusta ja esim. voitosta/häviöstä.

TD package:

Luokkarakenteet on luotu niin, että pelin osalla on itse tiedossa omat oleelliset tietonsa ja luokka Game yhdistää kaiken muun yhteen toimivaan kokonaisuuteen.
Enemy sisältää vihollisia käsittelevät tiedot. Metodeista tärkeimmät ovat *move* joka liikuttaa vihollisen sijaintia ja getHit, jolloin “hp” laskee.
Tower sisältää torneja vastaavat tiedot. Metodeista withinRange löytää kantamalta ensimmäisen vihollisen ja cooldown huolehtii hyökkäysten välisestä ajasta. towersToSaveState huolehtii IO-streamistä. wrap metodi etsii eri tornien tiedosita paremetrin mukaiset tiedot.
Wave on vihollis aaltoja kuvaava luokka. Se tietää eri aaltoja vastaavat viholliset ja siirtyy indekseillä eteenpäin.
Pos kuvaa yksinkertaisesti sijaintia, joka on torneilla, vihollisilla ja kentällä. Sillä on x ja y arvo sekä lisäykseen käytettävät metodit.
Overall kuvastaa yleistä tilannetta pelaajan omasta näkökulmasta. Tiedossa on mms. raha ja elämien tilanne ja niiden muutoksesta vastaavat metodit.
Numbers sisältää vain pari muuttujaa. Loin tämän kun yksittäisten arvojen vaihtamisesta moneen paikkaan kertyi liikaa vaivaa.
Level vastaa karttaa eli itse tasoa. Muuttuja road palauttaa tiedon vihollisten kulkureitistä. pointOnRoad kertoo onko cursorin piste tiellä (tornien asettamisessa oleellista) ja oob kertoo onko piste kartan ulkopuolella.
Game sisältää pelin kannalta olennaisten toimintojen metodit ja yhdistävät tekijät muiden luokkien kanssa yhdeksi kokonaisuudeksi ja IO-streamin kanssa toimimisen. Tornien asettelu ja vihollisten liikkuminen, sekä ajan kulku tapahtuu tässä luokassa.

SaverLoader, joka tallentaa/lataa pelin tietoja Json tiedostoon. Tässä huomioitavaa, että pelin muuttujat, kuten kyseinen aalto tai tornien paikat sijaitsevat tiedostossa save_state.json kun taas staattiset muuttujat kuten kartta sijaitsee conf.json tiedostossa.

Algoritmit:

Vihollisten kulkua kuvaava algoritmi toimii siten, että se tarkastelee reitin (level.road) pisteitä ja määrittelee nämä pisteet vihollisen kulkureitiksi. Vihollisen sijainnista tarkastetaan lähin piste, joka on reitillä ja tarkemmasta sijainnista grafiikoiden mukaisesti onko vihollinen millä puolella tätä pistettä. Tornien kantamasta riippuen lasketaan myös trigonometrian avulla onko tornin ympärille keskitetyn ympyrän sisällä vihollisia.

Tietorakenteet:

Sijainti käsitellään vain Pos luokan avulla, joka sisältää yksinkertaisuudessaan x ja y koordinaattien sijainnit. Suurin osa tiedoista on kiinni muuttujissa. Pelin kannalta muuttuvat tiedot ovat aika liukuvia kun taas grafiikat eivät muutu tai Json konfiguraatio, joka on ennen pelin alkua määritelty. Vihollisten ja tornien tiedot sijaitsevat buffereissa.

Tiedostot ja verkossa oleva tieto:

Tiedostoissa käsiteltävä tieto json tiedostoissa on todella yksinkertaista. Parametrejä voi helposti muuttaa haluamakseen, sekä save_state että conf tiedostoissa. Kartan muuttamisessa kannattaa huomioida graafikoiden muuttuminen erilaiseksi, mutta tätä voi helpottaa Numbers luokan parametreja vaihtamalla.

Testaus:

Projektia työstäessä en luonut erillisiä testiohjelmia. Sen sijaan suoritin erillisiä metodeita lisäten niihin println tai vastaavia komentoja, joiden avulla selvitin olivatko parametrit haluttuja. Graafikoiden kanssa testasin erilaisia tapoja lisätä omaa grafiikkaa Swingillä, mutta en saanut sitä toimimaan kunnolla ja tästä jäi surkastunut Pictures kansio, jota ei

lainkaan käytetä. Isoin debugattava ongelma oli kun enemy.move metodissa oli kaksi kertaa addx kutsuttu. Selvitin ongelman lähtemällä liikkeelle ylimmistä metodeista (game.moveEnemy) ja println() komennoilla selvitin askel kerrallaan ongelman alkuperän.

Ohjelman tunnetut puutteet ja viat:

Kartan kokoonpanosta riippuen viholliset kulkevat joskus väärään suuntaan oikoen vihreän halki. Viimeistely on kanssa jäänyt vajaaksi ja ajan puutteen takia piti jättää välistä mms. ammusten tekeminen. Ohjelmaa ei ole paljoa dokumentoitu myöskään ja kommentointia ei ole. Koodista on vaikea saada selvää jos ei ole hyvin perehtynyt eri asioihin ja vaikka nimeäminen on jotenkuten järkevää ei aina ole selvää mihin eri osat viittaavat.

3 parasta ja 3 heikointa kohtaa:

Hyviä asioita:
Ohjelman muokattavuus ja parametrien vaihto on todella helppoa.
Peli toimii yksinkertaisesta ja sen ymmärtäminen on helppoa.

Huonoja asioita:
Koodi ei ole kovin selkolukeista erityisesti isoissa luokissa GUI ja Game
Peli ei oikeastaan tarjoa haastetta ja on jokseenkin tylsä
Yksinkertaisuus lue myös tylsyyttä ja peliin kyllästyy helposti

Poikkeamat suunnitelmasta, toteutunut työjärjestely ja aikataulu:

Loin projektiin selkeän pohjan alunperin ja työstin sitä muutaman viikon aika laiskasti. Suunnitelmasta jäi puuttumaan pelin tyyli kokonaan ja

grafiikoista tuli yksinkertaisia muotoja. Luokkahierarkia muistuttaa kyllä teoriassa alkuperäistä suunnitelmaa. Joitan luokkia on vain lisätty helpottamaan selkeää työnjakoa eri osilla.
Aikataulusta käytin varmaan yhteensä 10 tuntia alkuperäisen pohjan tekemiseen ja viikkoa ennen palautusta poistin siitä melkein puolet. Käytin 4 päivänä noin 10 tuntia viimeisellä viikolla työ suorittamiseen tämän hetkiseen vaiheeseen. Otin paljon mallia verkosta löydetystä tornipuolustuspelistä ja kirjoitin sen pohjalta hyvin perusteelliset muistiinpanot työn toteutuksesta. Kirjoittamiseen meni paljon aikaa ja ongelmien ratkaisemiseen sekä debugaamiseen loput. Loin ensin konkreettiset luokat ja sen jälkeen graafiset luokat.

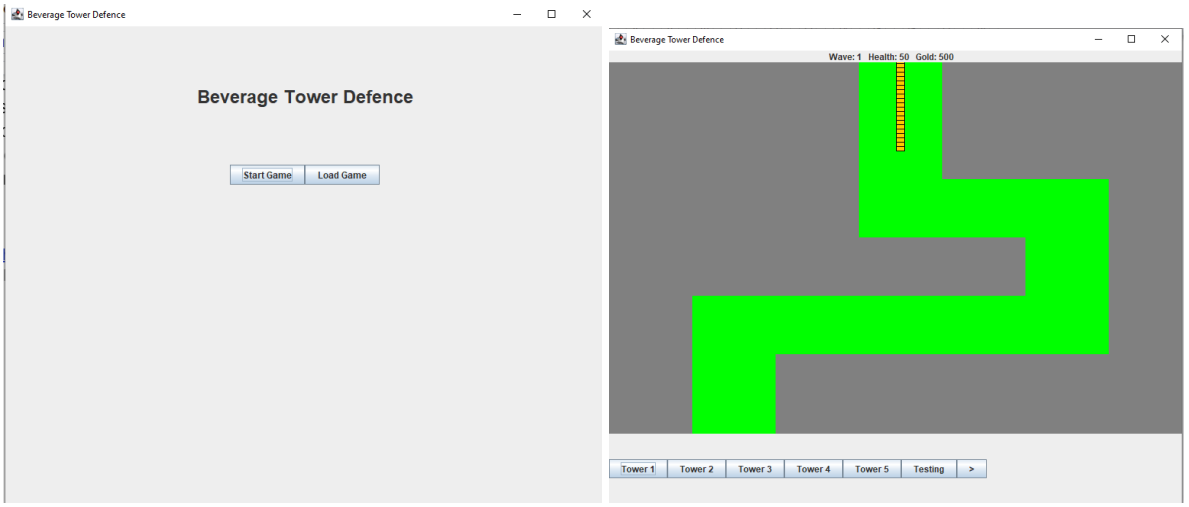
Kokonaisarvio lopputuloksesta:

Lopputuloksessa saavutin mielestäni pisteen, jossa ohjelma tekee sen mitä haluan. Pelinä ohjelma ei kuitenkaan ole kovin hyvä, mutta sen helppo muokattavuus vähentää sen painoarvoa. Grafiikoiden puute tekee myös pelistä hieman tylsän näköisen. Jos alkaisin uudestaan tekemään projektia varaisin sille huomattavasti enemmän aikaa ja kysyisin enemmän neuvoa. Debugaaminen oli todella aikaa syövää ja ajoittain hermostuttavaa ja jatkossa kunnioitan sitä varmasti enemmän.

Viitteet:

Osoitteesta: https://github.com/Desentso/TowerDefense löytyy peli, jonka pohjalta tein suunnitelman. Oma projektini rakenne muistuttaa sitä hyvin paljon.

Liitteet:



Lähdekoodi: https://version.aalto.fi/gitlab/tiusana1/at_towerdfence