jamk.fi

Loppuraportti

Markus Jylhänkangas
Pauli Kokkonen
Veeti Karttunen

Harjoitustyö
Huhtikuu/2016
Tekniikan ja liikenteen ala
Tieto- ja viestintätekniikan tutkinto-ohjelma

Jyväskylän ammattikorkeakoulu
JAMK University of Applied Sciences

Sisältö

1 Johdanto.....3

1.1 Rakennekaavio.....3

1.2 Pelin kuvaus.....3

1.3 Kohdeyleisö.....3

1.4 Käyttöympäristö ja käytetyt teknologiat.....3

1.5 Työnjako.....3

2 Luokat.....4

2.1 Game.....4

2.2 GameState.....4

2.2.1 PauseState.....4

2.2.2 MenuState.....4

2.2.3 GameOverState.....4

2.2.4 PlayState.....4

2.3 GameStateMachine.....5

2.4 InputHandler.....5

2.5 LoaderParams.....5

2.6 GameObject.....5

2.6.1 Enemy.....5

2.6.2 Player.....5

2.6.3 Bullet.....5

2.6.4 EnemySpawner.....5

2.6.5 MenuButton.....6

2.7 SoundManager.....6

2.8 TextureManager.....6

2.9 Vector2D.....6

3 Työaika.....6

4 Työtehtävät.....7

4.1 Yhteistyö.....7

4.2 Markus.....7

4.3 Veeti.....7

4.4 Pauli.....7

5 Kuvat ja ongelmat.....8

5.1 Ongelmat Collision systeemin kanssa.....8

6 Testaus.....8

7 Itsearviointi.....9

7.1 Markus Jylhänkangas.....9

7.2 Pauli Kokkonen.....9

7.3 Veeti Karttunen.....9

1 Johdanto

Github: https://github.com/Jylhis/Ghost-Spawner

1.1 Rakennekaavio

Linkki

UML kaavio sisältää projektissa käytetyt luokat. Kaavio sisältää todella monia luokkia, mutta useat niistä ovat melko samankaltaisia, kuten playstate, menustate, gameoverstate ja pausestate.

1.2 Pelin kuvaus

Peli sijoittuu huoneeseen jonne pelaaja on jostain kumman syystä joutunut, ja hänen täytyy selviytyä vihollisten hyökkäyksistä. Kyseessä on lintuperspektiivistä kuvattu areena peli. Viholliset tulevat pelimaailmaan aalloissa ja niiden määrä tulee lisääntymään mitä kauemmin pelaaja onnistuu selviämään. Pelimaailma koostuu huoneesta, pelaajasta ja vihollisista.

1.3 Kohdeyleisö

Pelin kohdeyleisö on henkilöt, jotka pitävät aalto pohjaisista taistelupeleistä. Peli on myös melko lailla ajantappoa varten, joten käytännössä kuka tahansa voi kiinnostua pelistä.

1.4 Käyttöympäristö ja käytetyt teknologiat

pelimoottorin luonti, sillä valitsemamme kirjasto ei sitä meille tarjoa. Kirjasto, jota käytämme on siis SDL2 niminen kirjasto, joka on äärimmäisen yksinkertainen kirjasto, joten kaikki on tehtävä itse, kuten grafiikan lataaminen, ikkunan luonti ja näppäinsyötteiden ottaminen.

Iso osa harjoitustyötä on myös pelimoottorin luonti, sillä valitsemamme kirjasto ei sitä meille tarjoa. Harjoitustyössä käytimme SDL2 (Simple DirectMedia Layer) kirjastoa grafiikan ja äänien esittämisessä sekä inputin kaappaamisessa. SDL2 kirjasto on äärimmäisen yksinkertainen kirjasto, joten kaikki on tehtävä itse, kuten grafiikan lataaminen, ikkunan luonti ja näppäinsyötteiden ottaminen.

1.5 Työnjako

Tähän mennessä kaikki mitä on saatu jo valmiiksi on lähes täysin tehty koulussa ja

yhdessä, vaikkakin joitain asioita ja luokkia on muokkailtu ja hiottu kotonakin.

Tulevaa työnjakoa on mietitty, että Markus hoitaisi kentän latauksen sekä sen luonnin. Veeti tekee vihollisen toiminnallisuutta, kuten tekoäly ja siihen liittyvät asiat. Pauli hoitaa damagen ja aseiden luonnin.

2 Luokat

2.1 Game

Game luokka hallitsee pelin toimintaa ja vastaa manageri-luokkien ja SDL:än käynnistymisestä. Game luokka käsittelee pelin sulkemisen.

2.2 GameState

Gamestate luokka on abstrakti luokka, joka toimii pohjana siitä periytyville pelitiloille.

2.2.1 PauseState

Luokka on aktiivinen pelin ollessa pysäytetty ja poistuessaan siirtyy playstate luokkaan.

2.2.2 MenuState

Menustate on valikkoa hallitseva tila. Menustatesta voi siirtyä playstateen tai sulkea pelin.

2.2.3 GameOverState

Tila johon siirrytään pelaajan kuoltua. Tästä voidaan aloittaa uusi peli tai siirtyä valikkoon.

2.2.4 PlayState

Tämä tila on itse peli. Tässä luokassa hallitaan kaikki törmäykset ja itse pelin logiikka.

2.3 GameStateMachine

Gamestatemachine on luokka, joka hallitsee pelitiloja ja niiden vaihtamista.

2.4 InputHandler

Inputhandler luokka hallitsee ohjainten toimintaa (hiiri, näppäimistö, peliohjain yms.).

2.5 LoaderParams

Loaderparams luokka pitää sisällään gameobjectien aloitusasetukset eli konstruktorin parametrit (koordinaatit ja objektin koko).

2.6 GameObject

Gameobject luokka on entiteettien yläluokka.

2.6.1 Enemy

Enemy on vihollisen entiteetti, jolla on satunnaisliikkuvuus tekoäly ja se voi vahingoittaa pelaajaa.

2.6.2 Player

Player eli pelaaja, on käyttäjän ohjaama luokka, joka voi ampua luoteja ja liikkua ympäriinsä.

2.6.3 Bullet

Luotiluokka, eli luotia kuvaava luokka, joka vahingoittaa vihollisia.

2.6.4 EnemySpawner

Tämä luokka synnyttää vihollisia kolmen sekunnin välein.

2.6.5 MenuButton

Buttonluokka, eli nappi. Tätä voidaan napauttaa hiirellä ja siitä tapahtuu toiminnallisuutta.

2.7 SoundManager

On luokka, joka lataa, hallitsee ja toistaa ääniä.

2.8 TextureManager

On luokka, joka lataa, hallitsee ja piirtää tekstuureita.

2.9 Vector2D

On vektoriluokka, eli vektori.

3 Työaika

Olemme työstäneet harjoitustyötä tasaisesti helmikuusta asti, joten tunteja on kertynyt melko paljon. Hiihtolomalla pidimme noin viikon mittaisen tauon projektin työstämisessä.

- Markus: noin 100 tuntia
- Pauli: noin 70 tuntia
- Veeti: noin 90 tuntia

4 Työtehtävät

4.1 Yhteistyö

Kaikki suuret luokat ja projektin rakenne on pohdittu yhteistyöllä. Suurimmat ongelmat korjattiin yhdessä. SDL kirjaston toiminnan aikaansaaminen tehtiin myös porukkalla.

4.2 Markus

Markus teki tilaluokat, äänten hallinta, törmäysten käsittelyä. Pieniä korjauksia bugien sattuessa.

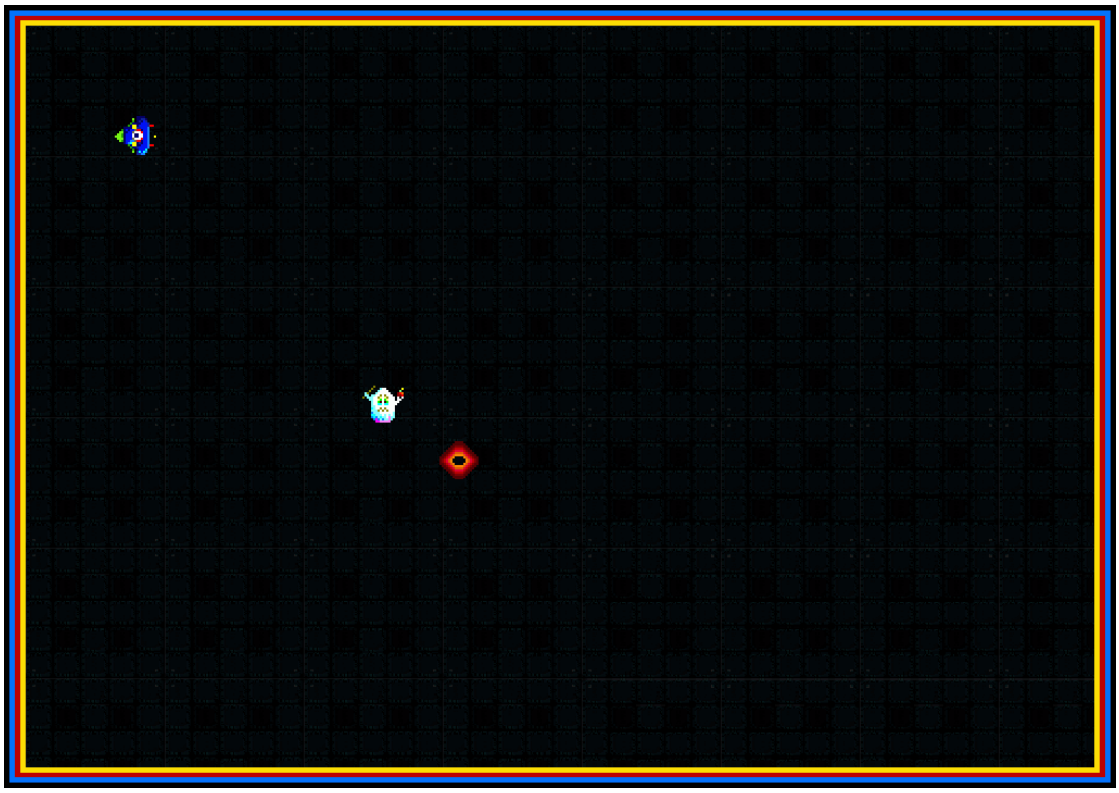
4.3 Veeti

Veeti työsti vihollista, luodin käyttäytymistä, pelaajan käyttäytymistä ja vektoreita. Pieniä korjauksia bugien sattuessa.

4.4 Pauli

Pauli teki pelin äänet, damagen, health, grafiikat. Pauli työsti myös tile pohjaista kentän käsittelyä, mutta osoittautuessaan liian monimutkaiseksi ja ongelmalliseksi se jätettiin pois. Pieniä korjauksia bugien sattuessa.

5 Kuvat ja ongelmat



5.1 Ongelmat Collision systeemin kanssa

Pelaajan ja vihollisen välillä oli ongelmia törmäyksen käsittelyssä, sillä diagonaalista törmäystä ei tunnistettu. Törmäyksen käsittely päätettiin korjata if else -lauseryppäällä.

6 Testaus

Peliä on testattu käytännössä eri työstämisvaiheiden välissä ja useita virheitä on löydetty testausten aikana, jotka on jälkeenpäin onnistuttu korjaamaan. Esimerkiksi viholliset liikkuivat jatkuvasti samaan suuntaan syntyessään, joka korjattiin luomalla satunnaisgeneraattori seed, joka perustuu useaan eri parametriin ja kellonaikaan.

7 Itsearviointi

7.1 Markus Jylhänkangas

Projektin työstäminen onnistui mainiosti. Haasteita on ollut riitävästi ja uutta asiaa tullut mukavasti. Olen oppinut huomattavasti olio-ohjelmoinnista tämän projektin aikana. Projektissa kaikki ryhmän jäsenet ovat olleet aktiivisia ja mukana tekemisessä. Arvosanaksi itselleni antaisin 5.

7.2 Pauli Kokkonen

Työnteko oli ihan hauskaa, tosin, joskus tuli kohtia, joissa ei tiennyt mitä tehdä. Työnjako oli meidän kohdalla selvä. Olen oppinut paljon Kurssin aikana, ja tiedän, että tulen tarvitsemaan kurssilla oppimiani asioita tulevaisuudessakin. Olen myös ylittänyt omat odotukseni. Arvioisin itseni 4.5/5.

7.3 Veeti Karttunen

Omasta mielestäni olen haastanut itseäni riittävästi tämän harjoitustyön aikana ja osallistunut työntekoon aktiivisesti. Ryhmämme kokoonpano on ollut toimiva, jokainen on ollut aktiivinen ja osallistunut työntekoon. Mielestäni olen ansainnut arvosanaksi 5 tästä kurssista.