Game of Blocks

TTOS0300 - Harjoitustyö

Lauri Virtanen

Opinnäytetyö

Huhtikuu 2017

Insinööri(AMK), Tieto- ja viestintätekniikk

Sisältö

[1 Johdanto 2](#_Toc480930451)

[1.1 Aikataulutus 2](#_Toc480930452)

[2 Ohjelman toiminnot 3](#_Toc480930453)

[2.1 Perustoiminnot 4](#_Toc480930454)

[2.2 Liikkuminen 5](#_Toc480930455)

[2.3 Hyppääminen 5](#_Toc480930456)

[2.4 Valikot 6](#_Toc480930457)

[3 Ongelmat 8](#_Toc480930458)

[3.1 Pelihahmon liikuttaminen 8](#_Toc480930459)

[3.2 Hiiren disablointi 9](#_Toc480930460)

[3.3 Vääränlainen alkusuunnittelu 9](#_Toc480930461)

[4 Testaus 9](#_Toc480930462)

[5 Pohdinta 9](#_Toc480930463)

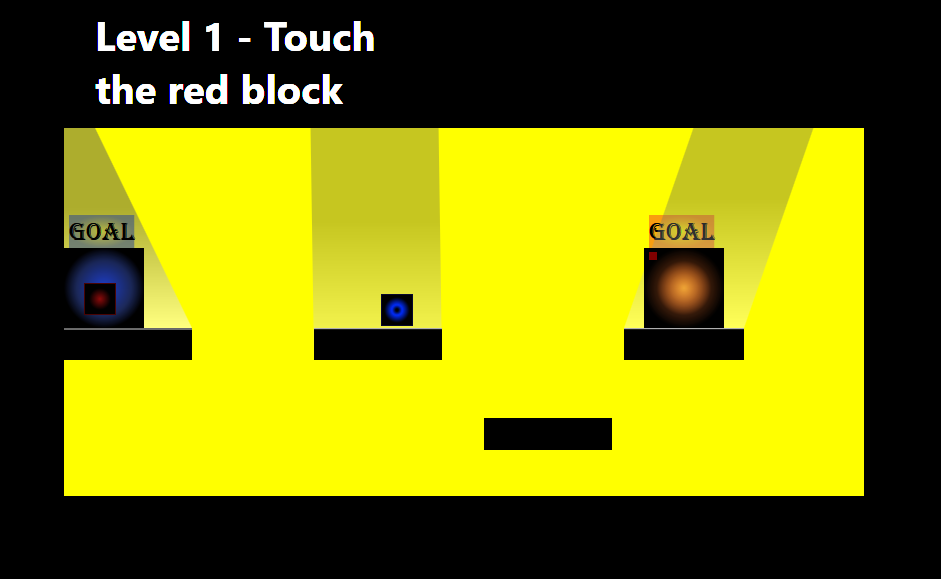
[5.1 Itsearviointi 10](#_Toc480930464)

[6 Kommentit 10](#_Toc480930465)

# Johdanto

Tämä on Lauri Virtasen K8455 oma, kurssin TTOS0300 loppuraportti, kurssilla oli tarkoituksena tehdä joko WPF tai UWP ohjelma.

Tein WPF sovelluksena pienen tasohyppely pelin, josta päädyin karsimaan erilaiset ansat ja tietokone vastustajat pois. Pelin tarkoituksena on liikkua sinisellä laatikolla niin, että saadaan liikutettua punainen laatikko punaiseen maaliin ja pelaaja siniseen maaliin.



Kuvio 1. Näyttökaappaus

## Aikataulutus

Harjoitustyön tekemiseen oli aikataulutettu 60 tuntia, joka tuli täyteen, jos lasketaan raportointi, suunnittelu ja yleinen logiikan pyörittely. Pelimekaniikat ja logiikat söivät paljon enemmän aikaa, kuin alun perin olin suunnitellut. Tämä siitä huolimatta, että yritin pitää pelin skaalan todella pienenä.

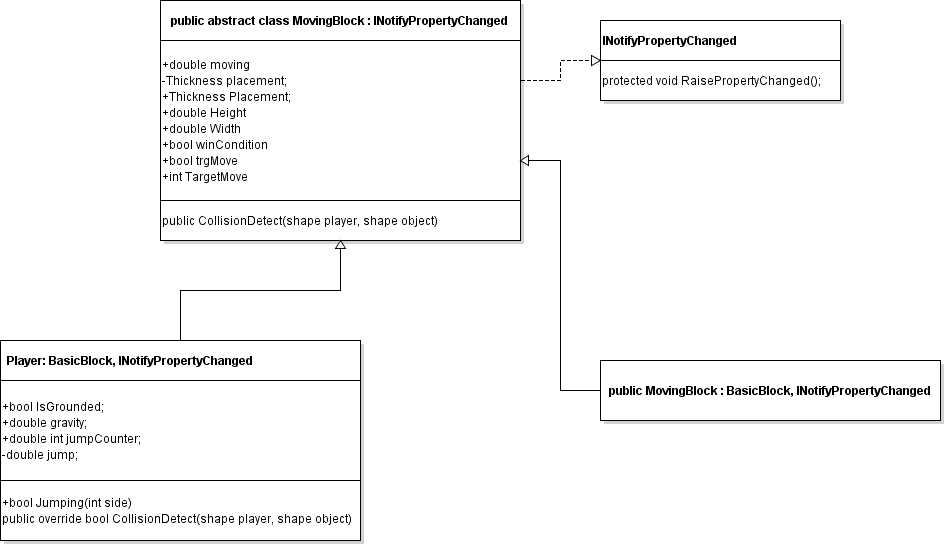
Aikataulutuksen suunnittelua ei kokonaisuudessaan ole tehty ennakkoon. Huomasin, että tein tai halusin tehdä tiettyjä asioita peliin, sitä kautta lisäsin niitä suunnitteluun.

|  |  |  |
| --- | --- | --- |
| Aihealue | Suun(h) | Tot.(h) |
| Aihe | 2 | 3 |
| Yleinen suunnitelma | 3 | 4 |
| UML ja luokat | 3 | 3 |
| Mockup | 2 | 2 |
| Olio - pelaajaluokka | 6 | 4 |
| Pelisuunnittelu | 8 | 10 |
| Pelimekaniikat | 8 | 8 |
| Kenttäsuunnittelu | 4 | 6 |
| Pelihahmon liikuttaminen | 4 | 6 |
| Pelin ulkonäkö | 3 | 4 |
| Äänet ja musiikki | 3 | 4 |
| Splashscreen | 1 | 1 |
| Koodin siistiminen | 10 | 4 |
| Koodin ja grafiikan yheistoiminta | 8 | 6 |
| Epätoivo | 0 | 3 |
| Paniikki | 0 | 3 |
| Yhteensä | 65 | 71 |

Aikataulua en aina muistanut pitää, joten joissain kohdissa voi olla liikaa ja joissain liian vähän. Itse pelilogiikan suunnittelua ja tekemistä voi olla reilusti yli lasketun sama kenttien suunnittelussa. Molemmissa aiemmin mainituista, suurin osa suunnitellusta ei päätynyt peliin tai totesin sen turhaksi asiaksi lisätä palautukseen mennessä.

# Ohjelman toiminnot

Kuviossa 2 olevien luokkien lisäksi on omat luokkansa niin vilkkuville valoille/väreille sekä Config luokka josta pystytään käyttämään sovelluksen sisällä yleisiä attribuutteja. Tämä mahdollisti lopussa retry napin toimimisen, kun olin muuttanut mihin kentät avataan alkuperäisen toteutuksen jälkeen.



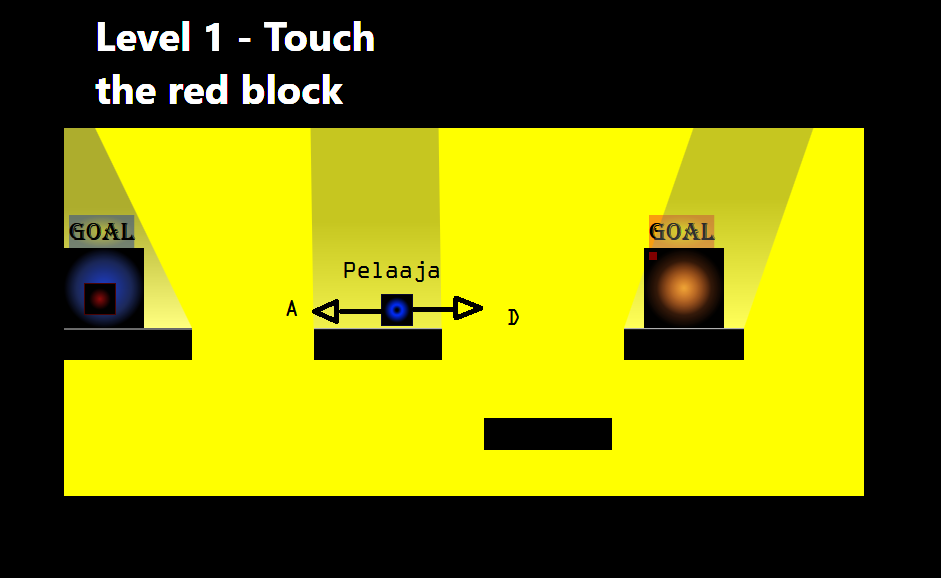
Kuvio 2. Luokkakaavio

## Perustoiminnot

Pelissä asiat tapahtuvat yhden DispatcherTimerin tickin aikana, jonka määrittelin 1 millisekuntiin. Tämän tickin aikana päivittyvän methodin sisälle lisäsin toisen try-catchin, joka heittää virheviestin MessageBoxiin.

Peliin lisätyt perustoiminnot ovat, pelaajahahmon liikuttaminen, törmäykset, punaisen palikan kanssa törmäys, taustamusiikki, iconit, pause, exit, retry.

## Liikkuminen



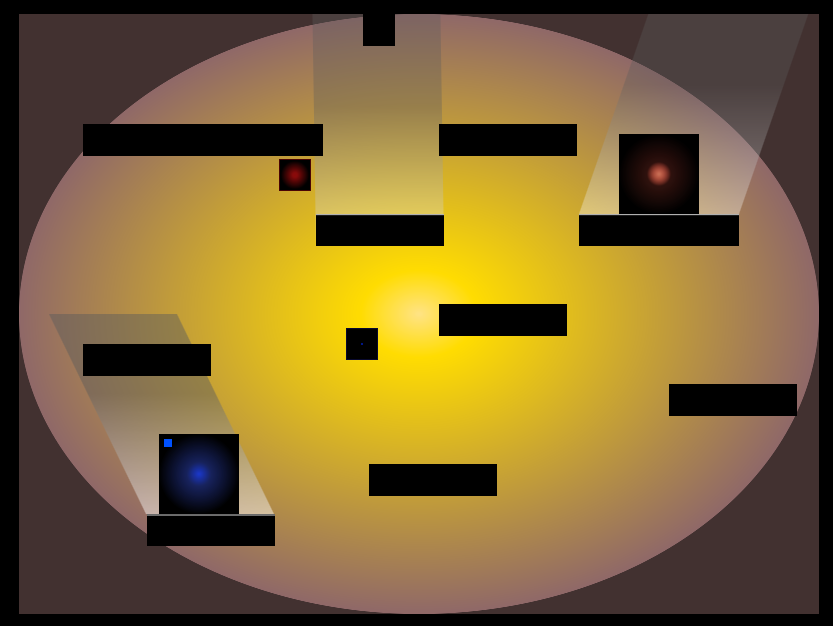
Kuvio 3. Liikkuminen perus näkymässä

Kuviosta kolme näkyy kaikki oleellinen pelistä. Sinisen sävyistä pienempää laatikkoa liikutetaan A ja D sekä space napeilla. A:sta menee vasemmalle ja D:stä menee oikealle ja spacesta pelaaja hyppää.

Liikkuminen on toteutettu lisäämällä tai poistamalla rectanglen marginista arvoja, aluksi toteutin saman asian niin, että se muuttaa vain koordinaatteja missä päin kanvaasia se on. Näin jälkikäteen ajateltuna, olisi ollut helpompi ja järkevämpi toteuttaa palikoiden asettelu muutoin kuin margineilla.

## Hyppääminen

Hyppy on mahdollistettu sillä, että pelaajaan kohdistuu koko ajan ”painovoima”, eli pelaaja liikkuu jatkuvasti yhden Dispatcher timerin tickin aikana kanvaasilla alaspäin. Hypyn korkeutta ja pituutta voidaan säätää pelaajaluokan sisältä, player luokasta löytyy yksinkertainen laskutoimitus, millä hypyn kaari on saatu järkeväksi.



Kuvio 4. 4. kenttä ja hypyn tarpeellisuus

## Valikot

Tein ohjelman alkuun pienen valikon, josta voidaan käynnistää itse peli. Sen graafiseen ilmeeseen tai ulkonäköön ei ole juuri muuten panostettu, kuin vaihdettu ikkunan otsikko, lisätty pelin iconi ja laitettu vähän eloa Play nappiin.



Kuvio 5. Alkuvalikko ennen peliä.

Tämän lisäksi pelissä on pause-valikko, johon pääsee painamalla esciä. Valikko ei täysin peitä pelin näkymää. Valikossa on kolme nappia, josta voidaan joko jatkaa peliä, aloittaa alusta tai lopettaa peli. Valikossa myös aina näkyy mitä nappeja pelissä käytetään liikkumiseen.



Kuvio 6. Pause valikko, joka tulee Esciä painamalla

# Ongelmat

Isoimmat ongelmakohdat olivat: Pelilogiikka, pelimekaniikkojen toiminta, WPF:n sivun muuttaminen ilman, ettei logiikkaa tarvitsee kopioida ja pelissä hiiren liikuttamisen estäminen.

## Pelihahmon liikuttaminen

Sain itse pelihahmon liikkumaan ja vertaamaan törmäyksiä suhteellisen helposti ja nopeasti, mutta sen jälkeen liikuteltavan ja pelaajalaatikon keskinäinen vertaus vertaus, jolla sai punaisen laatikon liikkumaan haluttuun suuntaan, osoittautui haastavaksi. Useaan kertaan luulin, että logiikat toimivat, mutta uusien asioiden lisäämisen jälkeen huomasin, että piti taas korjata aikaisempaa koodia.

## Hiiren disablointi

Ihmettelin todella kauan, miten WPF:stä saisi halutussa kohdassa estettyä hiiren toiminnot, koska kotikoneellani pelissä hiiren liikuttaminen jostain syystä hidasti pelin pysähtymiseen asti. Asia selvisi vasta viimeisellä viikolla, että se johtui omasta koneestani, koska peli liikkui normaaliin vauhtiin koulun koneella, vaikka hiirtä liikutti.

## Vääränlainen alkusuunnittelu

Ylivoimaisesti isoin ongelma oli se, etten ymmärtänyt olio pohjaista suunnittelua tarpeeksi hyvin ja en osannut aikataulun rajoissa muuttamaan kaikkea toimivaa luokkapohjaiseksi, vaikka olisin halunnut. Tämä ihmetytti itseäni, koska ymmärsin asiat olio-ohjelmoinnin kurssilla ja luulin, että niitä olisi yksinkertaista käyttää ja muuttaa jos ensin saa logiikat toimimaan ilman olioita.

MVVM- tavalla olisi pitänyt rakentaa tämä ohjelma. Alku suunnittelu ja lopputoteutus vaativat aivan liikaa kopioitua koodia muualta päin ohjelmaa. Loppu ajasta huomasin ja opin monia asioita millä peli olisi kannattanut toteuttaa, mutta en ollut tarpeeksi varma, että olisin saanut peliä toimivaan muotoon lopulliseen palautukseen.

# Testaus

Ohjelmaa on testattu pelaamalla sitä, melkein jokaisen muutoksen jälkeen. Sekä loppuaikoina koitettu toimiiko peli muillakin koneilla, josta selvisi uusia ongelmia tai ongelmien korjauksia.

Testaus tapoja en oikein keksinyt ja mitä pitäisi testata ja miten virheitä joissain tilanteissa tulisi analysoida.

# Pohdinta

Harjoitustyössä tuli todella paljon ongelmia, mitkä olivat lähtöisin aihevalinnasta. Pelin tekeminen sinänsä olisi voinut olla hyvä aihe kyseiselle kurssille, mutta se, että valitsee oman pelin mihin ei voinut tai halunnut kopioida mitään logiikoita tai koodeja, josta syystä peli ei voinut olla koskaan ”valmis”. Liiallinen ajankäyttö itse pelin hyvyyteen ja pelattavuuteen liittyen söivät aikaa pois itse WPF:n opiskelulta.

Kaikesta tästä riippumatta, minusta tuntuu, että opin todella paljon, miten peli olisi kannattanut tehdä. Alusta asti olisi pitänyt keskittyä MVVM tapaan ohjelmoida sekä hyvään luokka rakenteeseen, kentän vaihtomekaniikkaan, tallennukseen ja ylipäätään koodin tekemistä siihen pisteeseen, että olemassa olevia asioita olisi voinut lisätä yksinkertaisesti (esimerkiksi kentät tai uudet mekaniikat pelaajalle tai ansat ja viholliset) lisää.

## Itsearviointi

Todella vaikeaa antaa itsearviointia omasta tekemisestä, kun en ole yhtään tyytyväinen lopulliseen työhön. Tuntuu silti, että opin todella paljon ja osaisin tehdä paljon nopeammin ja paremmin seuraavan projektin.

Alkukurssista kuvittelin, että tavoittelisin vitosta tästä kurssista, mutta sen verran törmäsin ongelmiin mitä en aikataulun puitteissa saanut korjattua ja monet asiat eivät toimi haluamallani tavalla pelissä, jonka takia arvioisin tekemisiäni 2-3 arvoisesti. Suunnitelman mukainen peli, jossa toimii kentän vaihdot ja törmäyslogiikat jotka ovat perusta koko pelille.

# Kommentit

Hyvää kurssissa on se, että koko arvostelu tulee harjoitustyöstä, vaikkakaan en odota ihmeitä omastani.

Vaikea arvioida, mitä kaikkea harjoitustyöhön tulisi laittaa. Omasta työstäni puuttuu paljon hyvää, mutta sen tekemisen aikana oppi niin paljon, mitä ei kannata tehdä ja miten sitä ei kannata tehdä, mutta ei ollut aikaa tai järkeä ruveta lopussa vielä muuttamaan isoja virheitä ohjelmassa.

Mahdollisesti aikaisempien kurssien projekteja voitaisiin käydä läpi, myös koodillisesti. Tai voisit tehdä jonkun listan mitä ihmiset yleensä eivät tee oikein tai järkevästi.

Kokonaisuutena hyvä kurssi, olin tyytyväinen opetukseen ja materiaaleihin (vaikkakin googlen kautta 90% asiosita etsinkin).