

Peli: Tasohyppely - Projektisuunnitelma

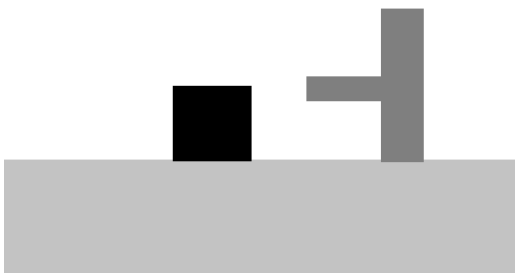
Tavoitteena on toteuttaa tasohyppely-peli, jossa pelaaja ohjaa hahmoa kentän läpi hyppimällä erilaisten tasojen päälle, kuten esimerkiksi pelissä Donkey Kong. GUI:n kehittämiseen on käytössä kirjastot PyQt5 tai vaihtoehtoisesti PySide library. Tavoitteenani on toteuttaa keskivaikkea vaikeusaste, ja sen jälkeen mahdollisesti vaativa, jos aikaa ja voimia riittää.

Käyttötapauskuvaus ja käyttöliittymän luonnos

Ohjelman kommunikointi käyttäjän kanssa perustuu graafiseen käyttöliittymään, jossa on mahdollisesti painike pelin aloittamiseen sekä jos on enemmän kuin yksi kartta vaihtoehto niin kartan valitsemiseen oma painike. Graafisen käyttöliittymän kautta pelaaja näkee oman peli hahmonsa suhteessa muuhun peliympäristöön, sekä pystyy vaikuttamaan oman hahmonsa liikkeisiin painamalla näppäimiä näppäimistössään. Eli ohjelma saa syötteensä pelaajan hiiren painalluksista sekä näppäimistön painalluksista. Aloitusruudulla todennäköisesti tulee olemaan hyvin yksinkertainen painike, joka sanoo "Start Game", sekä kartta valikossa, jos sellainen tulee olemaan, on eri painikkeet eri kartoille, jotka ovat nimettyjä karttojen mukaisesti. Mahdollisesti aloitusnäytöllä sekä itse pelin aikana tulee olemaan näkyvillä "top score" tai jonkin näköinen tulostaulu. Samoin peli voi kysyä pelaajan hahmolle nimen syötteenä. Itse pelin ohjelmaruudulla tulee näkymään pelaajan objekti jossain kiinteässä pisteessä ruudulla. Pelaaja objektin ympäröivä maailma "liikkuu" pelaaja objektin suhteen. Itse maailmassa olevia esteitä, mahdollisia NPC (non-player character) olioita, maailma rakennetta tai voittamiskehoja en ole vielä päättänyt. Tästä huolimatta liikkumisen yksi varma muoto on pelaajan objektin hyppiminen.

Kuvitelma miltä peli voisi näyttää:

TOP SCORE: 420

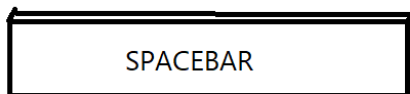


Esimerkki käyttötapaus ohjelmalle

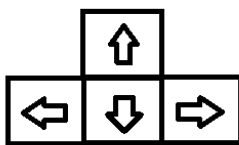
Ohjelman ja käyttäjän rajapintana toimii ohjelmaikkuna, jossa on graafisesti näkyvillä käyttäjälle mahdolliset toiminnot tai pelimaailma, jossa pelaaja voi liikutella omaa hahmoaan. Ohjelma käynnistetään ajamalla sen main-funktio jonkin python compilerin kautta, jolloin ohjelma aukaisee ikkunan, jonka kautta pelaaja on vuorovaikutuksessa ohjelman kanssa. Jos pelissä on vain yksi kartta, niin ohjelman aloitusikkunassa on vain todennäköisesti painike "Start Game", jota painamalla peli käynnistyy. Tämän jälkeen pelaaja voi syöttää omalle pelihahmolle nimen, jota voi käyttää esimerkiksi pistetaulussa. Kun käyttäjä painaa tätä painiketta, niin peli luo kartan ja pelimaailman objektit sekä näyttää sen graafisena käyttäjälle. Samoin jos pistetaulu on käytössä, niin ohjelma luo uudelle pelaajalle oman kohdan pistelistalle. Pelin kartalle sekä pelimaailman objekteille on omat oliot, jotka luovat pelimaailman ns. aloitusmuotoon. Kun kartta ja pelin objektit on luotu, niin voi pelaaja näppäimistönsä nappeja painamalla liikuttaa hahmoa pelimaailmassa. Pelaaja siten voi antamalla syötettä ohjelmalle tavoitella ns. voittoehtoja. Samoin pelimaailmassa on ehdot häviämisellä, jolloin pelaaja voi halutessaan painaa painiketta, jolla hän pääsee kokeilemaan uudestaan. Mikäli jos pelaaja haluaa sulkea pelin, niin hän voi painaa ikkunan yläkulmassa olevaa rastia, jolloin ohjelma sulkeutuu.

Mahdolliset pelihahmon liikuttamiseen käytettävät näppäimet:

Hyppiminen:

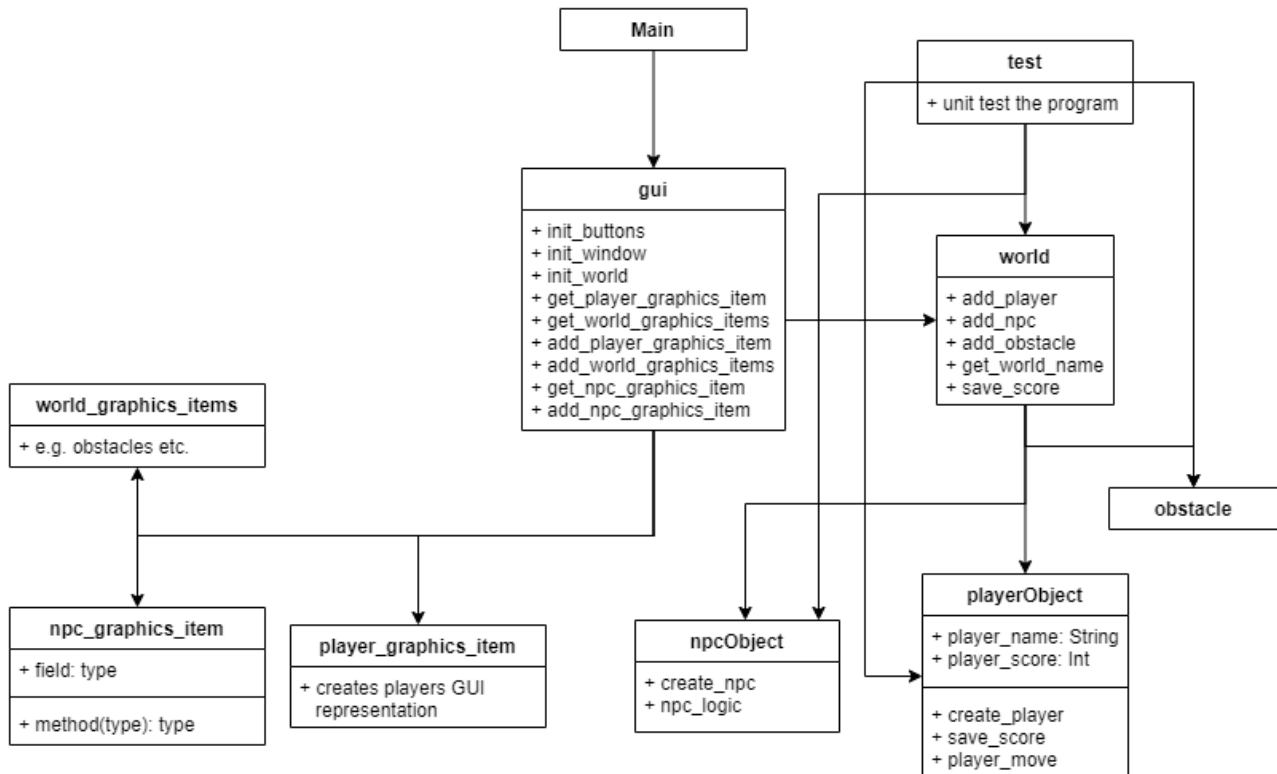


Liikkuminen:



Projektin rakennesuunnitelma

Ohjelma tarvitsee osia, jotka luovat kaikki graafisen käyttöliittymän osat, kuten pelin ikkunan, pelin hahmot sekä pelimaailman. Samoin peli tarvitsee osan, joka luo kaikki tarvittavat objektit kuten pelimaailman ja lisää sinne kaikki hahmot (pelaajan hahmo ja mahdolliset NPC-hahmot ja esteet). Samoin peli tarvitsee main-funktion, josta peli ajetaan sekä test-funktion, josta pelin eri osia voi testata. Tein pelin rakenteesta hyvin yksinkertaisen ja pintaraapaisu UML:n:



Tietorakenteet

Ohjelman tietojen varastointiin soveltuvat parhaiten todennäköisesti muuttuvan kokoiset rakenteet sekä taulukot. Esimerkiksi muuttuvan kokoisia rakenteita voisi käyttää, jos on pistetaulukko käytössä tai jos tarvitsee lukea dataa samalla kun peli etenee. Samoin jos on joitain kiinteitä rakenteita pelissä niin niitä voidaan tallentaa taulukoihin, jos se on sopiva ratkaisu tilanteeseen. Kuitenkin en ole täysin varma oman ohjelman tietorakenteesta vielä.

Tiedostot ja tiedostoformaattit

Pelin tallennustiedostot, jos niitä tulee olemaan, ovat todennäköisesti jonkinlaisessa tekstitiedostossa (.txt) muodossa, ja siitä voidaan lukea kaikki tarpeellinen tieto. Peliin saatan ottaa käyttöön jonkinlaisia grafiikka elementtejä pelihahmoille sekä pelimaailmalle (mahdollinen sivusto, josta lataan grafiikkapaketit: <https://opengameart.org/>). Paketit ladattaisiin ohjelmaan todennäköisesti PNG-muodossa.

Algoritmit

Pelaajan liikkumiseen pätee jonkin näköiset pelimaailman fysiikan lait, kuten jos pelin hahmo hyppää niin saa se kiihtyvyyttä alaspäin, sekä hahmo voi törmätä pelimaailman objekteihin kuten seiniin. Samoin pelaaja voi seistä tasoilla ilman että putoaa niistä läpi. Algoritmien pitäisi myös pystyä muuttamaan näppäimistön syöte hahmon liikkeeksi tai toiminnaksi. Samoin mikäli jos pelimaailmassa tulee olemaan NPC-hahmoja, niin niille pitäisi implementoida "tekoäly", joka ohjaa niiden toimintaa pelimaailmassa. Pelissä pitäisi olla näiden lisäksi voitto ja/tai häviö ehdot ja kerättyjen pisteiden lasku.

Testaussuunnitelma

Ohjelmassa on hyvä testata eri objektien luominen sekä lisääminen, siten että varmistuu objektien oikeanlaisuudesta. Samoin olisi hyvä testata eri grafiikka elementtien luominen ja lisääminen, sekä graafisen käyttöliittymän oikeanlaisuuden. Näiden lisäksi, jos pelissä on mahdollisuus tallentaa ja ladata tallennus niin näiden oikeanlaisuus ja toimivuus olisi hyvä testata, sekä jos peliin ladataan grafiikka elementtejä esimerkiksi pelin hahmoille, niin nämä olisi syytä testata. Näppäimistö syötteiden toimivuus on tärkeä testata, samoin pelata itse peliä eri tavoin löytääkseen mahdollisesti piileviä bugeja.

Kirjastot ja muut työkalut

Ohjelmassa on käytössä GUI:n luomiseen joko PyQt5 tai PySide kirjastot. Nämä kirjastot ovat olennaisia, jos haluaa luoda graafisen käyttöliittymän ja graafisia objekteja.

Aikataulu

- | | |
|-------------|--|
| 26.2 | - Projektisuunnitelman palautus |
| 5.3 | - Ohjaustapaaminen assarin kanssa |
| 14.3 | - Projektin ohjelmointi aloitettu |
| 21.3 | - Perus rakenne hahmottunut sekä perusasiat implementoitu |
| 26.3 | - Checkpoint pidetty |
| 4.4 | - Projektin ohjelmointia sekä perus gui-elementit implementoitu sekä pelimaailman objektien koodaaminen aloitettu |
| 18.4 | - Projektista jonkinlainen testaus valmis versio sekä kaikki tarvittavat elementit ohjelmoitu, testaaminen aloitettu |
| 25.4 | - Projektin viimeistely ja loppuhioaminen |
| 7.5 | - Projektin palautus deadline klo 14:00, koodit palautettuna Gitiin |

Kirjallisuusviitteet ja linkit

Graafisia paketteja: <https://opengameart.org/>

Esimerkki pelin koodaamiseen PyQt5:lla:

<https://gist.github.com/rogerallen/f06ba704ce3befb5459239e3fdf842c7>

PyQt5 tutorial: <https://zetcode.com/gui/pyqt5/>

PyQt5 hyppiminen: <https://www.geeksforgeeks.org/making-label-jump-using-pyqt5-in-python/>

PyQt5 liikkuminen: <https://stackoverflow.com/questions/57238032/how-do-i-make-this-pyqt5-character-move-on-arrow-key-input>