

Patrik Ahola 591111

Sähkötekniikan kandidaatti 3. vuosi

04.03.2020

Yleinen kuvaus projektista

Toteutan pythonilla yksinkertaisen tasohyppelypelin, jossa Super Mario tyyliin liikutaan vasemmalta oikealle niin että vasemmalle voi palata vain sen matkan joka kyseisellä hetkellä on renderöity. Pelissä kuitenkin maa on laavaa, joten siihen koskeminen aiheuttaa välittömän kuoleman. Peli koostuu hahmosta, jota pelaaja liikuttaa, tasoista joiden avulla pelaajan on tarkoitus liikkua ja vihollisista jotka yrittävät eri tavoin tuhota pelaajan. Tavoite on päästä oikealle maaliin asti hengissä. Pelaajan liikuttamalla hahmolla on oletuksena jonkinlainen lyömäase mutta tasoilta on mahdollista löytää parempia aseita jotka kuitenkin toimivat rajatun ajan tai niissä on rajattu määrä ammuksia. Pelin häviää jos vihollinen tuhoaa pelaajan tai pelaaja tippuu laavaan x-määrän.

UI

Ohjelma avaa käynnistettäessä yksinkertaisen valikon, jossa on kaksi klikattavaa nappia "start" sekä "exit". Käyttäjä valitsee näistä toisen klikkaamalla sitä hiirellä. Napit toteutetaan PyQt-kirjaston button-widgeteillä joihin yhdistetään halutut metodit pelin aloittamiseksi tai sen sulkemiseksi. Itse pelissä liikkuminen tapahtuu nuolinäppäimillä ja space-barilla. Ohjelma tunnistaa näppäinten painamisen PyQt:n keyPressEventin avulla. Start napin painalluksella peli alkaa eli ruudulle ilmestyy taso sen kaikkine elementteineen. Ruudulla näkyy lisäksi elämien määrä sekä sen hetkinen "health". Peli tulostaa tarvittaessa viestit ruudulle hetkeksi aikaa kunnes viesti katoaa.

Tiedostot ja niiden formaatit

Pelissä esiintyvät ainakin jossain määrin satunnaisesti generoitavat objektit eli viholliset , aseet, ammukset, (alusla luotava pelaajan hahmo) jne. luodaan tekstitiedostojen pohjalta.

Tekstitiedoston formaatti voisi esimerkiksi olla seuraava:

-Tyyppi :(hahmo/ase/projektiili/physicalObj)

-Ulkonäkö : tiedostossa sijaitsevan kuvan polku

-Koko : x, y

Kukin ominaisuus sijoittuu tekstitiedostossa omalle rivilleen. Pelkän tyypin perusteella voidaan jo koodissa luoda objektille oma logiikka tai rooli eli valita sille oikea luokka. Ulkonäkö sekä koko määräytyvät kahden viimeisimmän kentän perusteella.

Testaus

Niin kuin ohjeistuksessakin mainittiin, on näin alkuvaiheessa projektia vaikea laatia täydellistä testikirjoa. Kuitenkin ohjelman toimintaa pääosin voidaan testata esim. unitestien avulla. Ensisijaisesti haluamme tietenkin testata että peli käynnistyy start:ia painamalla ja sulkeutuu exit:iä painamalla ja että näiden painaminen on ylipäättän mahdollista. Kun tämän todetaan toimivan niin haluamme varmistaa pelin logiikan ja fysiikan olevan kunnossa. Eli testaamme pelaajan hahmon reagoivan nuolinäppäinten sekä space-bar:in painallukseen. Testaamme myös pelaajan ottavan vahinkoa vihollisista, aseista, projektiileista sekä laavasta ja myös totta kai vihollisten tekevän samoin. Myös pelin loppuminen elämien loppuessa tai maaliin saavuttaessa testataan. Näillä testaamme jo seuraavat asiat.

- Pelin voi aloittaa

- Pelissä pelaaja voi liikkua

- Pelaaja ja viholliset reagoivat halutulla tavalla haluttuihin objekteihin

- Peli loppuu kun sen pitää loppua

Totta kai projektin edetessä ja erityisesti sen loppuvaiheessa yksityiskohtaisempia testejä eri skenaarioihin tehdään, mutta yllä mainitut testit kattavat ainakin alkuvaiheessa pääpiirteittäin ohjelman toiminnan.