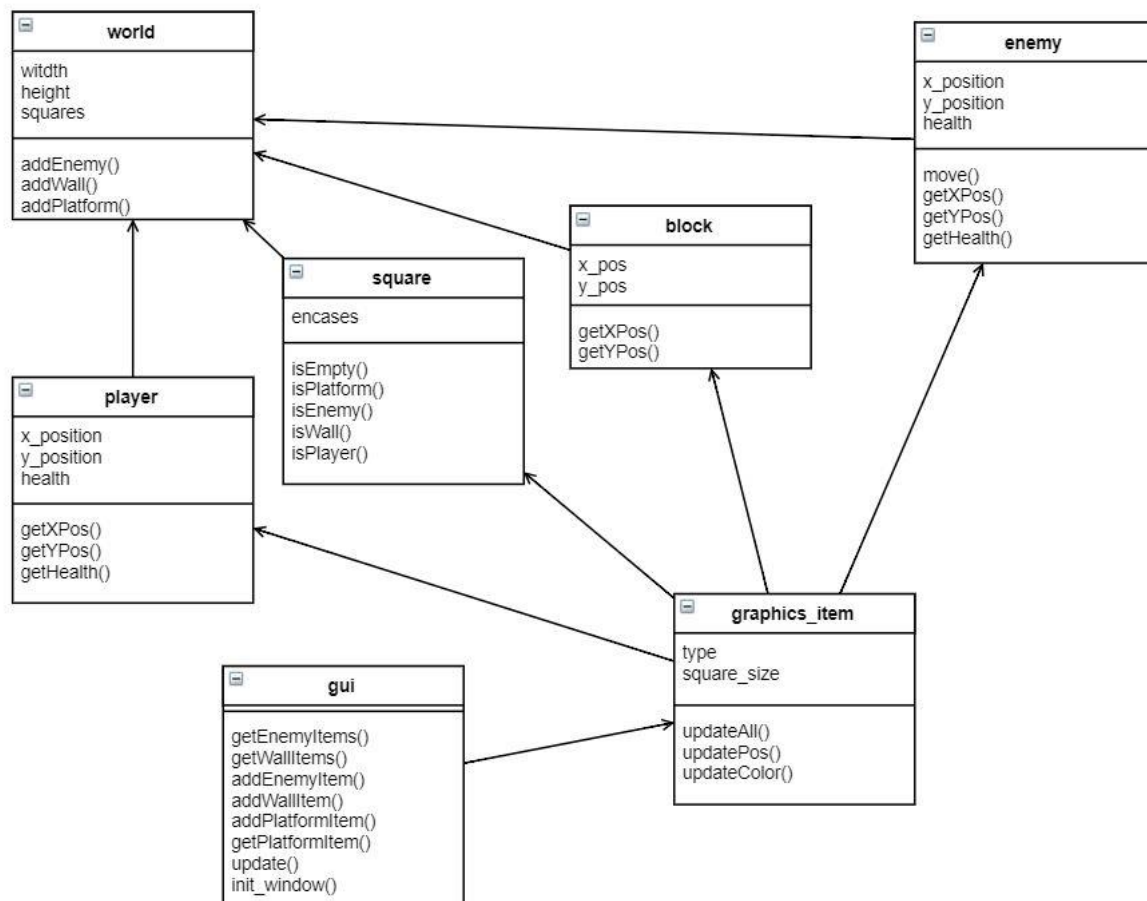


Tekninen suunnitelma

1. Rakennesuunnitelma

Kuvassa esitelty luokkarakenne UML-tapaisessa muodossa:



Tarkoituksena on mukailla robotworld-tehtävän luokkarakennetta, kuitenkin siten, että se sopii oma projektiini toteutukseen. Kuvassa esitelty myös keskeiset metodit ja muuttujat.

2. Käyttötapakuvaus

Ensimmäisenä, kun käyttäjä avaa ohjelmani, hän näkee valikon, josta voi valita Help tai Play. Jos käyttäjä valitsee Help, hän pääsee näkymään, jossa selitetään, miten peli toimii ja kuinka pelaajaa liikutetaan. Jos taas hän valitsee Play, pääsee hän suoraan pelaamaan peliä.

Pelissä pelaajaa liikutetaan nuolinäppäimillä oikealle ja vasemmalle ja välilyöntinäppäimestä hypätään. Käytännössä kuitenkin pelaaja on paikallaan ja ympärillä olevat tasanteet ja seinät liikkuvat sitä mukaa kun käyttäjä haluaa liikkua pelissä. Viholliset liikkuvat suhteessa tasanteisiin ja seiniin oman mielensä mukaisesti.

Jos käyttäjä ohjaa pelaajan ulos ruudulta (eli esim. rotkoon) tai törmää viholliseen, tulee ruudulle ilmoitus: Game Over ja vaihtoehdot Play Again tai Menu. Play Again käynnistää pelin alusta ja Menu ohjaa käyttäjän aloitusnäkyyn.

3. Algoritmi t

Tarkoituksena on luoda simppeli pelimoottori, joka päivittää käyttäjän syötteestä ja vihollisten etenemisestä johtuvaa liikettä ruudulla. Lisäksi simppeli törmäyksen tunnistin seuraa, mikäli pelaaja on törmännyt esimerkiksi viholliseen tai muuhun kiellettyyn.

4. Tietorakenteet

Tarkoitukseni on käyttää pythonin valmiiksi rakennettuja tietorakenteita pelin sisällä tapahtuvien tilanteiden tallentamiseen. Esimerkiksi world luokka tallentaa pelikentän ruudukon python listaan ja graphics_item tallentaa type-muuttujaansa pythonin API luokan enum tyyppisen arvon, joka liittyy joko viholliseen, rakenteeseen, pelaajaan tai tyhjyyteen.

5. Aikataulu

Viikko 1: Tavoite saada tärkeimpien luokkien toteutus kuntoon. Tunnit: 4h

Viikko 2: Tärkeimpien luokkien hiomista ja vähemmän tärkeiden luokkien hahmottelua ja testejä. Tunnit: 6h

Viikko 3: Tavoitteena saada simppeli pelin toteutus tähän mennessä valmiiksi (pelaaja liikkuu maailmassa) ja testejä. Tunnit: 6h

Viikko 4: Lisää ominaisuuksia edellisen viikon toteutuksen päälle (esim. kuoleminen...). Tunnit: 6h

Viikko 5: Käyttöliittymän tekemistä ja pelin hiomista. Tunnit: 6h

Viikko 6: Viimeisiä silauksia ulkonäköön ja pelattavuuteen: Tunnit: 7h

Viikko 7: Viimeistely. Tunnit: 6h

6. Yksikkötestaussuunnitelma

Keskeisiin yksikkötesteihin lukeutuu ainakin pelaajan paikan testaus, suhteessa muhin pelimaailman objekteihin ja ruudukoissa liikkuvien objektien oikea liike suhteessa pelaajaan. Ensimmäistä tapausta on tarkoitus testata miten pelaaja reagoi, jos esim. seinä tulee vastaan tai tiputaan tasanteelta. Toisessa tapauksessa testataan, että pelimaailman kaikki osat liikkuvat oikein suhteessa toisiinsa, eli esim jos käyttäjä haluaa että pelaaja siirtyy yhden ruudun oikealle, kaikki "blockit" siirtyvät yhden ruudun vasemmalle.

7. Kirjallisuusviitteet ja linkit

pyQt: <http://pyqt.sourceforge.net/Docs/PyQt5/>

Python documentation: <https://docs.python.org/3/>