tiistai 23. helmikuu 2021

Projektisuunnitelma – Golden Retriever

Henkilötiedot

Tekijä:

Olli Suoniemi 657 330

Insinööritieteiden kandidaattiohjelma, vuosikurssi 2018

Yleiskuvaus

Tämän projektin tarkoituksena on harjoitella ohjelmointiprojektin tekemistä ja sen hallintaa tekemällä 2d-videopeli Python kielellä käyttäen ulkoista PyQt -kirjastoa. Tähän kuuluu itsenäisen ohjelman suunnittelu ja toteutus. Suunnittelussa käytän apunani UML-luokkadiagrammia havainnollistamaan ohjelmarakennetta ja luokkien välisiä yhteyksiä. Tavoitteenani on tehdä vaativampi versio ohjelmastani.

Pelin päähahmo on koira, joka etsii kultaista luuta. Tavoitteena on löytää kultainen luu tekemällä tunneleita ja kaivamalla maata. Pelin kuluessa, pelaaja voi löytää maasta erilaisia esineitä, jotka edesauttavat kaivamisprosessia. Pelaajaa voi löytää esimerkiksi erilaisia lapioita, jotka nopeuttavat kaivamista. Pelin tallentaminen tapahtuu löytämällä tietynlaisia pisteitä, jotka merkkaamalla peli tallentuu. Lisäksi kerätyllä maalla ja mineraaleilla voi ostaa kissakauppiaalta erilaisia tavaroita.

Käyttötapakuvaus

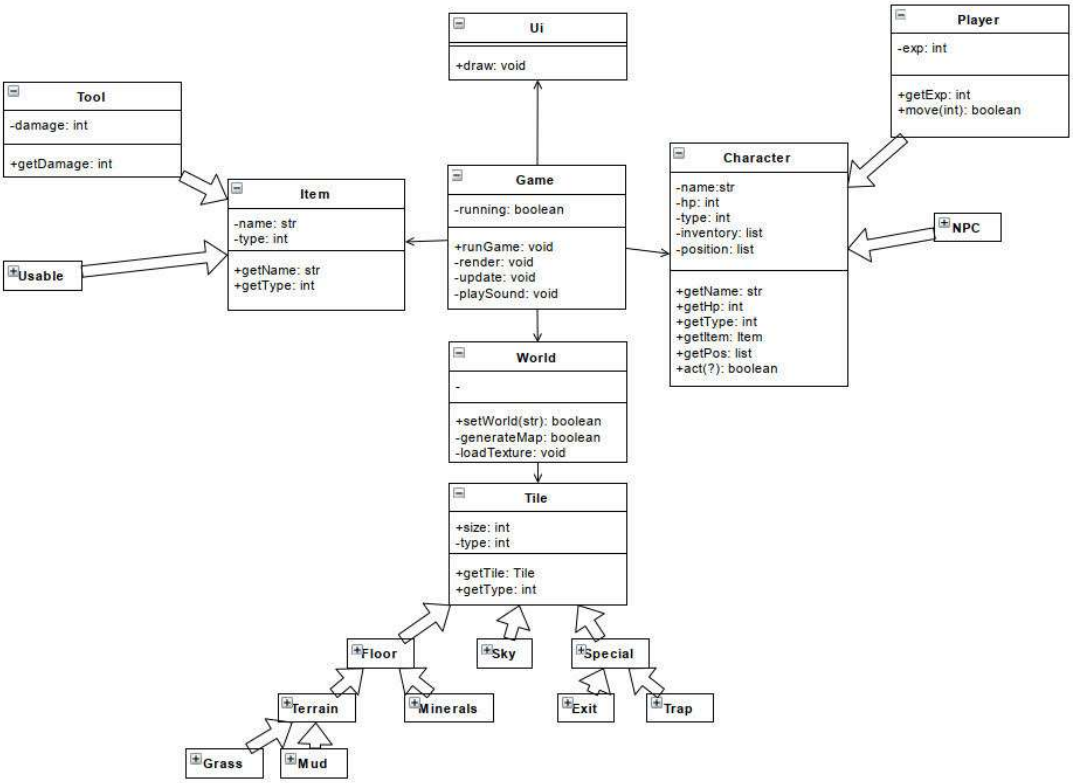
Ohjelmassa pelimoottorin virkana toimii silmukka, joka käsittelee pelin tapahtumat ja käyttäjän antamat syötteet. Peliä ohjataan perinteisesti näppäimistöllä, jonka mukaan pelin päähahmo liikkuu ja tekee toimintoja. Lisäksi hiirellä ohjataan mahdollisia valikkoja ja inventaarioita. Hiiren ja näppäimistön lukemisen lisäksi tarkoitus on lisätä myös ohjaimelle tuki. Peliä ohjaavat spesifit komennot päätetään myöhemmin.

Ohjelman käynnistyessä avautuu peli-ikkuna, josta käyttäjä voi valita uuden pelin aloittamisen, vanhan pelin lataamisen, ohjelman sulkemisen tai pelin tulostaulukon avaamisen, jossa näkyy peliä pelanneiden pisteet suurusjärjestyksessä.

Esimerkki ohjelman tyypillisestä käyttötavasta on seuraavanlainen; käyttäjä käynnistää ohjelman, jolloin ohjelman pääfunktio kutsuu peliluokkaa Game, joka kutsuu luokka Ui. Ui hoitaa käyttöliittymän piirtymisen sekä HUDin jossa näkyy pelin aikana pelaajalle tärkeitä tietoja, kuten pelaajan energiamäärän ja kerätyt pisteet.

Käyttäjä valitsee uuden pelin aloittamisen, jonka jälkeen ohjelma kysyy käyttäjältä nimeä. Käyttäjä syöttää nimikenttään nimensä, ja peli alkaa. Game kutsuu luokkaa World, jossa alustetaan ja ladataan maa. Game kutsuu pelin aikana tarvittaessa eri luokkia, kuten hahmo- ja esineluokkia. Käyttäjän saavuttaessa pelin lopun, ohjelma palaa takaisin peliluokkaan, joka hoitaa ohjelman lopetuksen.

Rakennesuunnitelma



Ohjelma koostuu main-luokasta, joka kutsuu ohjelman käynnistyessä peliluokkaa Game. Game-luokka toimii pelin sydämenä ja pelimoottorina, joka organisoi pelin tapahtumat, piirtämisen, pelitilan päivittämisen sekä pelin kuvantamisen eli renderöinnin. Ui:ssa rakennettaan käyttöliittymä, joka mahdollistaa pelin avaamisen ja pelaamisen. Ui hoitaa pelin aikana HUDin piirtämisen. Game luokan metodit update ja render ovat metodeita, joita kutsutaan metodista runGame jatkuvasti. Metodit update ja render eivät näy luokan Game ulkopuolelle.

World-luokassa alustetaan ja piirrettään pelimaailma sekä ladataan tekstuurit maailman piirtämistä varten. Metodi setWorld saa parametrinaan ladattavan maailman nimen. Metodi lataa maailman tiedostosta, ja palauttaa True jos se onnistuu. Muussa tapauksessa metodi palauttaa False.

Pelin maailma koostuu useista yksittäisistä Tile-luokan perijöistä. Luokka Tile on abstrakti luokka, jonka perii luokat Floor, Sky ja Special. Tile:n muuttuja size kertoo kuinka suuria yksittäiset Tilet ovat. Special-luokan erikoisuuksia on Exit ja Trap, joilla on erikoisempia ominaisuuksia, kuten hahmoon vaikuttaminen tai kentän vaihdon aikaansaaminen.

Character-luokalla on ilmiselviä muuttujia, kuten nimi ja hp. Luokka pitää myös sisällään hahmon sijainnin sekä tyypin. Hahmotyypillä on oma tyyppinumeronsa, jonka avulla ne saadaan identifioitua. Hahmolla on myös oma varasto, johon lisätään Item-luokan instansseja. Metodin act tarkempi määrittely tehdään myöhemmin. Metodi act kuvaa tässä suunnitelmassa kaikkia niitä muita kuin suoraviivaiseen liikkumiseen liittyviä toimintoja, kuten esimerkiksi Item-instanssin käyttämistä. Player-luokan metodi *move* ottaa parametrikseen joko oikean (1) tai vasemman (-1), ja liikuttaa hahmoa sinne suuntaan. Metodi palauttaa boolean arvon kertoakseen, onko liikkuminen mahdollista.

Item-luokka kuvaa pelin käytettäviä esineitä. Näihin kuuluu muun muassa työkalut, erilaiset spesiaalijuomat ja mahdolliset aseet. Kuten Character-luokan tapaan, myös Item-luokassa jokainen instanssi on yksilöity type-muuttujalla.

Tietorakenteet

Käytän ohjelmassani Pythonin valmiita tietorakenteita. Listaa (list) käytän projektissani todennäköisesti tietorakenteista eniten. Esimerkiksi pelaajalle kuuluvat Item-objektit säilön pelaajalle kuuluvaan listaan. Käytän dynaamista rakennetta, koska pelaajalla voi olla mielivaltainen määrä Item-objekteja.

Tiedostot ja tiedostoformaatit

Pelin kenttä on tallennettu tekstitiedostossa, jossa jokaisella merkillä on merkityksensä. Jokainen kirjain tai numero vastaa jotain tiettyä Tile-luokan instanssia, jolla on jokin tietty tekstuuri. Tekstuurit tulen tekemään itse piirtämällä. Tekstuurit ladataan jpg-tiedostona kenttää generoitaessa. Tässä esimerkki havainnollistamaan, minkälaisessa formaatissa kenttä tallennettuna voisi olla.

xxxxxxxxxxxxxxxxxxxxxxxxxxx00000000xxxxxxxxxxxxxxx0000000000000000000xxxxxxxxxxxxx

xxxxxxxxxxxxxxxxxxxxxxxxxxx000000000xxxxxxxxxxxxxx000000000S000xxxxxxxxxxxxxxxxxxx

xxxxxxxxxxxxxxxxxxxxxxxxxxx1111S1111111xxxxxxxxx11111111111111111111111111xxxxxx

xxxxxxxxxxxxxxxxxxxxxxxxx1111111111111111111111111111111111111111xxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxx222222222222S222222xxxxxxxxxxxxxxxxxxx2222222222222222xxxx
xxxxxxxxxxxxxxxxxxx22222222222222222xxxxx22222222222222222222222xxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxx2222222222222222xxxxxxxxxxxxxxxxxxxxx22222222222xxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxx3333333333xxxxxxxxxxxxxxxxxxxxxx33333333333333xxxxxxxxxxxxxx
xxxxxxxxxxxxxx333333333xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx3333333333xxxxxxxxxxxxx
xxxx333333333333xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx3333xxxxxxxxxxxxxxxx
xxxxxxxxxxxxxx3333333xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

X = läpäisemätön seinä

0 = normaalia maata

1 = yhden asteen verran vaikeampikulkuista maata, jonka kaivamisesta hyötyy enemmän

2 = kahden asteen verran vaikeampikulkuista maata, jonka kaivamisesta hyötyy enemmän

jne..

S = erikoistiili, joka tekee jotain erityistä

Peliä tallennettaessa, tallennettaan generoidun kentän lisäksi pelin tiedot. Kentän tiedoston nimi ladataan seuraavalla kerralla sillä perusteella, kuin se on #MAPFILENAME-lohkossa tallennettu.

Pelaajatallennusten formaatti:

SAVEFILE

#DATE 23/02/2021

#CLOCK 13:34

#MAPFILENAME map1

#PLAYERNAME Olli

#HP 100

#EXP 16

#INVENTORY shovel10/sword/potion/potion/potion

SAVEFILE

Tallennustiedosto alkaa aina SAVEFILE-tekstillä ja päättyy SAVEFILE-tekstiin. Jokainen lohko alkaa #-merkillä ja päättyy seuraavaan #-merkin tullessa vastaan. Lohkojen järjestyksellä ei ole väliä. Lataamisen jälkeen ohjelma tarkistaa onnistuiko tallennuksen lataus, ja se, onko ladatut tiedot valideja.

Algoritmit

Pelin pääkartta on puolisatunnaisesti generoitu siten, että harvinaisen maan ilmenevyyden todennäköisyys kasvaa syvemmälle mentäessä ja siten, että yleisempää maata ei voi olla harvinaisemman maan alapuolella tai samalla tasolla eriarvoisen maan kanssa.

Kentän varsinainen generointi tapahtuu kentän maksimileveys- ja syvyysparametrien (jotka päätetään myöhemmin) rajoissa seuraavasti:

- Generointi alkaa 0-tasosta.
- Aukon on oltava vähintään 2 tiiltä leveä.
- Yhdellä rivillä voi olla maksimissaan 5 aukkoa.
- Seuraavan rivin aukon on generoituva sellaiseen kohtaan, että siitä on yhteys ylempään aukkoon.
- Harvinaisempaa maata ei voi generoitua yleisemmän yläpuolelle.
- Samalle riville voi generoitua vain samanarvoista maata.

Testaussuunnitelma

Projektissani tärkeä testauskohde on pääkentän oikeellisuus. Ohjelman täytyy pystyä generoimaan validi kenttä, jotta pelaaminen on mahdollista. Yksinkertaistaen, testatessa kenttää tarkistetaan, että sen luomisessa käytettävät ehdot täyttyvät.

Lisäksi ohjelman tulee pystyä lataamaan tiedostosta monia asioita, kuten tekstuurit, tallennustiedot ja musiikit. Näiden lataamisen onnistuminen tarkistetaan heti lataamisen jälkeen.

Tallennustiedoston lataamisessa tulee tarkistaa, että pelaaja luodaan, ja se, että pelaajalle asetetaan oikeanlaisia arvoja. Pelaajan nimi ei esimerkiksi voi olla mielivaltaisen pitkä, eikä siinä saa numeroita.

Kirjastot ja muut työkalut

Projektissani käytän PyQt5-kirjastoa pelin grafiikoiden ja mekaniikoiden toteuttamiseen sekä äänien linkittämiseen.

Tekstuurien piirtämiseen käytän Piskel-ohjelmaa, joka on helppokäyttöinen ilmainen ”sprite-editor”.

Aikataulu

Suunnitelmaan käytetty aika: 10 h

PyQt-kirjastoon tutustumiseen käytetty aika: 10 h

Kenttägeneraattoriin käytetty aika: 10 h

Luokkien ja metodien luomiseen käytetty aika: 20 h

Käyttöliittymään käytetty aika: 10 h

Pelin tallennusformaatin käytetty aika: 15 h

Tekstuurien piirtämiseen käytetty aika: 15 h

Testaamiseen käytetty aika: 10 h

Kokonaisaika: 100 h

vko 8	Suunnitelman tekeminen ja palautus. PyQt-kirjastoon tutustuminen.
vko 9	Kenttägeneraattorin luominen, peliluokan tekemistä. **SUUNITELMATAPAAMINEN**
vko 10	Kenttägeneraattorin ja peliluokan työstämistä
vko 11	Luokkien ja metodien sekä käyttöliittymän luontia.
vko 12	Luokkien ja metodien sekä käyttöliittymän luontia. **CHECKPOINT-TAPAAMINEN**
vko 13	Pelin tallennusformaatin luontia. Tallennusformaatin lukijan rakentaminen.
vko 14	Tekstuurien piirtämistä.
vko 15	Tekstuurien piirtämistä. Musiikin tekemistä/linkittämistä.
vko 16	Testaamista. Musiikin tekemistä/linkittämistä.
vko 17	Testaamista. Projekti valmis
vko 18	**PROJEKTI-DEADLINE**

Kirjallisuusvinkit ja linkit

https://docs.python.org/3/tutorial/datastructures.html Käytetty tarkistamaan Pythonin valmiiden tietorakenteiden ominaisuuksia

https://www.riverbankcomputing.com/static/Docs/PyQt5/ PyQt:n luojan RiverBank Computing Limitedin oma virallinen ohjekirja PyQt5:n käyttöön

https://wiki.python.org/moin/PyQt/Tutorials PyQt tutoriaaleja

https://www.piskelapp.com/ Piskel, ilmainen tekstuurinteko-ohjelma