

1. Henkilötiedot

Otsikko; tekijän nimi, opiskelijanumero, koulutusohjelma, päiväys.

Rosanna Manninen

779234

Bioinformaatioteknologia

3.5.2020

2. Yleiskuvaus

Oma projektini on strategiapeli. Pelissä on tarkoitus voittaa hirviö, jonka hp näkyy vasemmassa ala-laatikossa. Pelaajalla on kiltalaisia, jotka tulee pitää hengissä. Pelin voittaa tuhoamalla hirviön kiltiksellä tehtävien toimintojen avulla.

Pysyin aika hyvin alkuperäisessä suunnitelmassa pieniä muutoksia lukuunottamatta.

3. Käyttöohje

Ohjelma käynnistetään ajamalla tiedosto main.py. Peli alkaa kun painaa Uusi peli -nappia. Pelin pelaamisen löytyvät myös napin "Ohje" takaa. Myös pelin lore eli taustatarina on luettavissa. Pelistä poistutaan napin "Poistu" kautta tai ihan vain perinteisesti rastista.

Kiltiksellä voi: Lukea Snapsia, keittää kahvia, harjoittaa mindfulnessia, houkutella lisää kiltalaisia ja käydä kaupassa. Jokaisella toiminnolla on useita eri lopputuloksia, esim kaupasta voidaan tuoda S'napseja tai kahvia, vain jompaakumpaa tai ei kumpaakaan.

4. Ulkoiset kirjastot

Projektissa on käytetty kirjastoja PyQt5 ja moduulia sys, muuten koodi on kokonaan ihan perus pythonia. PyQt5 kirjastolla rakensin ohjelman graafisen käyttöliittymän.

5. Ohjelman rakenne

Ohjelman luokat:

Kiltis: Luokka, joka pitää pelin kulusta, toiminnoista ja muuttujista huolen. Sisältää hirviön toimintoja, pelaajan toimintoja, vähän pelaajan ja ohjelman välistä viestintää sekä algoritmipuolta. Kiltiksen toimintoja kutsutaan graafisen käyttöliittymän painikkeiden avulla.

- Pelaajan toiminnot kuten: lue_snapsi(), kauppa()...
- vuoroloppu(), tarkistaa tapahtuiko pelaajan vuoron aikana mitään, antaa hirviön tehdä vuoronsa ja tarkistaa sen seuraukset ja lopulta päivittää kierroksittain muuttuvan lämpötilan ja sen vaikutukset. Kiltiksen muuttujat tarkistavat kiltismuuttujat() ja muuttujienvaiikutukset().
- Hirviön vuoro: hirvioturn(), hirviön vuoro, jolloin se tekee vuorostaan jotakin, yleensä pahaa.
- Algoritmeja: muuta_luku(), enitenrp() yms..auttavat valitsemaan eri toimintojen kohteet ja mitä tapahtuu
- Pelin kontrollointia: peli_pois(), pelitarkistus()
- Viestintä: tilaviestim(), funktio uuden tilaviestin lisäämiseksi. Lisäksi muuttujaviesti() ja kiltalaisinfo(), jotka antavat gui:lle tarvittavat tiedot pelin kulusta.
- Iso kasa get-funktioita, joita hyödynnetty vähän siellä sun täällä.

Window: Luokka, joka toimii graafisen käyttöliittymän pohjana, perii luokan QMainWindow. Sisältää aika paljon kaikkea, lähinnä nappeihin ja labeleihin liittyviä funktioita, sekä päivitys funktio, joka toimii timerin avustuksella. Kommunikoi jatkuvasti kiltiksen kanssa.

Kiltalainen: Luokka kiltalaisille. Hahmo, jolla on nimi, rp, tyyppi ja kiltis, jolle se on sijoitettu. Funktiot lähinnä palauttavat ominaisuuksia, vahingoittavat tai parantavat.

Hirvio: Luokka hirviölle, jolla on nimi, hp ja tieto kiltiksestaan. Parannus ja vahingoitus funktioita. main.py ohjelman ajamiseen!

6. Algoritmit

Pelistäni en ole käyttänyt seed-pohjaista satunnaisuutta. Tästä syystä varianssia eri toimintoihin on pitänyt luoda muilla keinoilla. Ohjelma tarkkailee pelissä olevia muuttujia ja hahmoja jatkuvasti ja nämä arvot muuttuvatkin alati. Perustin pelin ja sen vastustajan toiminnan näihin muuttujiin.

Vastustajan toimintaan tarvittiin jonkin sortin tekoäly, joten toteutin sellaisen niin, että aina kun on hirviön vuoro sillä on viisi vaihtoehtoa. Jokaisella vuorolla lasketaan luku seuraavasti: kiltalaisten määrä + lämpötila + hirviön hp. Tästä luvusta selvitin tekijöiden määrän. Suuruusluokassa, jossa näillä muuttujilla tässä pelissä liikutaan on maksimissaan viisi. Ja tekijöiden määrän avulla hirviö ns. valitsee mitä tekee.

Myös kaikissa sekä hirviö, että pelaajan toiminnoissa on luotu varianssia jokaisen toiminto kerran välille. Se, miten hyvin jokin toiminto onnistuu, riippuu myös näistä muuttujista, esim. jonkin tietyn muuttujan parillisuus. Lisäksi joissain hirviön sekä pelaajan toiminnoissa olen joko suhteuttanut vahingon/kohteiden määrän kiltalaisten määrään sillä hetkellä.

Tämän kaiken olisi voinut tehdä vaikka pythonin random.randint(a,b) funktion avulla, niin että olisi laittanut vaikkapa ajan seediksi, mutta tällöin esim. pelin balansointiin olisi vaikeampaa vaikuttaa. Nyt tietyillä hirviö elämä pisteillä, jotkin asiat eivät onnistu tai ovat voimakkaampia/heikompia. Ongelman löytyessä tavallani saman komentopolun pystyi myös ajamaan uudestaan samanlaisena.

7. Tietorakenteet

Ohjelman tietorakenteissa ei mitään kovin erikoista.

8. Tiedostot

Ei käsittele tiedostoja

9. Testaus

Yksi olennainen osa pelin testausta oli ihan vain sen pelaaminen, laitoin ohjelman tulostamaan käyttämäni toiminnon nimen, jolloin sain selville tapahtumaketjun, jolla päädyttiin ohjelman kaatumiseen. Tämän perusteella debuggasin kyseisen ketjun. Koska koodin "satunnaisuus" ei perustu esimerkiksi seedin kautta käytettäviin random-funktioihin on ketjut mahdollista ajaa täysin samanlaisina.

Testauksen isoin haaste oli varmaankin mahdollisimman ison tilanne kirjon kattaminen, sillä pelin toiminnot riippuvat muuttujista.

Testasin kaikkia funktioita erikseen, eri tilanteissa. yksittäiset funktiot ja toiminnot ovat kuitenkin suhteellisen yksinkertaisia testata, joten tämä ei aiheuttanut kovin suurta päänvaivaa. Testasin kyllä vähän vähemmän kuin mitä suunnittelin.

Gui:ta testailin ihan vain ajamalla ja rakentelemalla siinä samalla.

10. Ohjelman tunnetut puutteet ja viat

Ohjelman gui:ssa olisi jonkin verran kehittämistä, mm. layoutteja hyödyntämällä ohjelmasta olisi varmasti saanut sopivamman eri alustoille.

Pelin kaikkien mahdollisten polkujen ja tapahtuma ketjujen testaaminen on käytännössä kyllä suhteellisen mahdotonta, sillä pelillä ei ole vakio kestoja ja muuttujia on paljon. Joten varmasti on yksittäisiä tapauksia joissa peli kaatuu löytyy todennäköisesti.

Pelin rakenteesta sen verran, että voisin erikseen luoda luokan nykyiselle kiltikselle, jolloin aina uuden pelin tapauksessa olisi mahdollista luoda uusi kiltis peliä varten. Tämä toki käyttäisi

enemmän muistia, mutta voisi mahdollisesti luoda lisää vaihtoehtoja siihen, mitä peliin voisi vielä lisätä.

11. 3 parasta ja 3 heikointa kohtaa

3 parasta:

Peli on mielestäni ihan huolellisesti ja ajatuksella tehty sekä toimii hyvin! Kiltis-luokan muuttajat ovat omasta mielestäni ja testauksen perusteella aika luotettavia tilanteessa kuin tilanteessa!

Olen omasta mielestäni saanut peliin ihan hyvää varianssia ja valinnanvaraa pelaajan kannalta. Pelissä eri muuttajat myös tuovat vaihtelevat olot.

Peli on helppokäyttöinen ja käyttäjäystävällinen!

3 heikointa:

Gui:ssa ns. chatin toteutus. Tähän olisin voinut käyttää esim. dialogi toimintoa tai messageboxia, mutten halunnut mitään ylimääräisiä ponnahdusikkunoita. Mutta toisaalta ehkä jopa hyväksi puoleksi voisi listata, että sain viestit peliltä ihan järkevän näköiseksi pelkän QLabelin avulla ja listalla, joka säilyttää viisi viimeisintä viestiä.

Muuttujapohjaiset algoritmit ja tekoäly voivat olla hankalia jos peliä lähtee muokkaamaan myöhemmin, jolloin myös ne pitäisi sovittaa uudestaan!

12. Poikkeamat suunnitelmasta

Pelin rakenne hieman eroaa siitä mitä olin suunnitellut, muttei ihan älyttömästi! Vähän mentiin yksinkertaisemman suuntaan.

Luokkajaossa jätin luokan Kiltalainen alaluokat pois ja tein kiltalaisten tyyppiin vain muuttujilla.

Suunnittelemani aikataulu ei pitänyt kyllä yhtään, missä kyllä itselläni voisi olla petraamista!

13. Toteutunut työjärjestys ja aikataulu

Projektin tein aikalailla kolmessa palassa, ensimmäinen osuus maaliskuun lopussa sain projektin ihan hyvin alkuun, lähinnä luokkakajoa ja koodille perusrakennetta. Tämän jälkeen oli checkpoint, jonka jälkeen huhtikuun alussa sain projektia ja lähinnä graafista käyttöliittymää jatkettua hyvin. Parin viikon tauon jälkeen viimeistelin projektin loput funktiot ja ominaisuudet ja korjailin virheitä (huhtikuun loppu/toukokuun alku).

14. Arvio lopputuloksesta

Omasta mielestäni projektista tuli ihan kiva ja pelattavissa oleva peli. Toimii nopeasti ja pitäisi olla ihan käyttäjäystävällinenkin. Pelin funktiot toimivat sujuvasti ja ilman ongelmia. Tein omaan silmääni aika siistiä ja helposti ymmärrettävää jälkeä. Joitakin asioita olisin rakenteessa ja pelin toiminnassa voinut tehdä toisin, mutta lähinnä projektin viime tippaan jättämisen vuoksi en jaksanut enää lähteä uutta kehittämään.

Lisäksi joita pelissä mielelläni myöhemmin näkisin olisivat ainakin se, että ohjelman käyttäjä voisi nimetä itse omat kiltalaisensa, nätimpi grafiikkapuoli sekä pelin edetessä uusien ominaisuuksien aukeaminen. Tällä tarkoitan sitä, että osa toiminnoista olisi käytettävissä vasta pelin myöhemmissä vaiheissa, sekä yhdellä vuorolla voisi tehdä useamman toiminnon tai valita kiltalaisen itse, joka suorittaa toiminnon.

Lisäksi luokalle kiltalainen voisin toteuttaa alkuperäisen suunnitelman alaluokat, jolloin voisi lisätä eroja eri kiltalaistyyppien toiminnallisuuksissa.

Jatkokehityksen kannalta pelin toimintaa pitäisi ehkä vähän muuttaa. Varsinkin jos haluaisi lisää kiltalaisille ominaisuuksia, mutta tällaisenaankin mm. kierrosten lukumäärään liittyvät muutokset ovat ihan mahdollisia.

15. Viitteet

Youtube-kanavan Tech with Tim lukuisat PyQt5:n käyttöön liittyvät videot.

https://www.youtube.com/channel/UC4JX40jDee_tINbkjycV4Sg

A+ materiaali, kursseilta Y1 ja Y2

16. Liitteet

Kuvia:

Ensimmäisessä ohjelma kun se on juuri käynnistetty. Peli ei ole vielä päällä, mutta sen voi käynnistää napista Uusi peli. Ohjeet ja lore ovat luettavissa niille kuuluvien nappien avulla. Lisäksi jokaiselle toiminnolle on oma ToolTip, joten tieto pelin toimimisesta on helposti saatavilla.

Poistua voi napista Poistu tai raksin kautta, jolloin ohjelma kysyy haluatko varmasti poistua.

Toinen kuva: peli on juuri aloitettu. Kiltiksellä on uudet kiltalaiset, aloitus resurssit ja hirviö. Ennen uuden pelin aloittamista toimintonappeja ei ole voinut käyttää.

Kolmas kuva: peliä on pelattu hieman! Kiltalaiset vaihtuvat, kierroksia kulunut jonkin verran. Tässä näkyy myös tuon viestihomman toiminta, viisi viimeisintä ilmoitusta näkyvät, uusin alinpana.



