

Y2 Projekti: Labyrintti

Nimi: Verner Turunen

Opiskelijanumero: 649403

Koulutusohjelma: ISM, Aalto yliopiston kauppakorkeakoulu, kandidaatin tutkinto

Vuosikurssi: 2017

Tekninen suunnitelma

1. Ohjelman rakennesuunnitelma

Ohjelmani luokkia tulevat olemaan ainakin:

1. Labyrinth(): luo labyrinttiobjektin joko ennalta määrätyn tai käyttäjän antaman koon mukaan. Luo ensin labyrintin Square-objekteista, joissa on kaikki seinät, jonka jälkeen algoritmin avulla luo weave-labyrintin, ja sijoittaa hiiren labyrintin keskelle.
2. Square(): labyrintti muodostuu square-objekteista, joilla on seiniä, joiden läpi hiiri ei pääse kulkemaan. Square-luokassa on metodi, jonka avulla saadaan selville hiiren sijainti, joten se on myös helppo tallentaa.
3. Direction(): määrittelee hiiren mahdolliset kulkusuunnat (weave labyrintissa neljä ilmansuuntaa)
4. Mouse(): luo hiiriolion, joka sijoitetaan luotuun labyrinttiin, ja jota käyttäjä komennoillaan siirtelee squaresta toiseen.
5. Lisäksi mahdolliset graafiset toteutukset, esim. luokat labyrinth_graphics () ja mouse_graphics ().

2. Käyttötapauskuvaus

Käyttäjä käynnistää ohjelman, ja saa eteensä merkkipohjaisen/graafigisen käyttöliittymän. Hän aloittaa hiiren ohjailun labyrintissa (määritetyillä komennoilla tai mahdollisesti nuolinäppäimillä). Käyttäjä haluaa jatkaa pelaamista myöhemmin, joten hän tallentaa pelin. Myöhemmin hän avaa tiedoston ohjelmassa, jolloin peli jatkuu samasta tilanteesta. Labyrintti osoittautuu liian vaikeaksi, jolloin käyttäjä napauttaa forfeit-nappia: ohjelma piirtää ratkaisun käyttöliittymään ja ohjaa hiiren ulos labyrintista.

3. Algoritmit

Labyrintin toteuttamiseen löytyi monia vaihtoehtoisia algoritmeja, joista päädyin valitsemaan ”recursive backtrackerin”. Kyseinen algoritmi käyttää labyrintin luomiseen ”passage carver” – metodia, jossa labyrintin kulkureitti muodostetaan alussa umpinaisiksi luoduista neliöistä. Se soveltuu hyvin weave-tyypin labyrintin luomiseen: se tapahtuu lähes samalla tavoin kuin normaalin labyrintin luominen, mutta weaven kanssa on huomioitava mahdolliset kulkureitit muiden kulkureittien yli ja ali, ja huolehdittava, että yleissuunnitelmassani mainitut ehdot umpikujista ja käännöksistä muiden reittien alla tai päällä toteutuvat.

4. Tietorakenteet

Ohjelmassa tarvittavan tiedon käsittelyyn soveltuu luultavasti parhaiten listat. Ne mahdollistavat erikokoisten labyrinttien tallennuksen ja käsittelyn. Algoritmin toteutusta varten tarvitsen myös listoja, johon tallennan labyrintin sisältämät Square-objektit, ja joita läpi käymällä toteutan weave-labyrintin tuottavan algoritmin.

5. Kirjastot

Standardikirjaston lisäksi tulen mahdollisessa graafisessa toteutuksessa käyttämään PyQt käyttöliittymäkirjastoa.

6. Aikataulu

Aluksi aion vielä tehdä lisää taustatyötä algoritmiin ja sen toteutukseen liittyen, tähän uskon menevän ainakin n. 5 tuntia. Luokkia lähden toteuttamaan pian ja yritän hahmotella enemmän lopullisia metodeja. Olen joskus hidas etenemään, joten tuntimääräisen arvion antaminen on hyvin vaikeaa. Kun pääsen vauhtiin, luulen kuitenkin eteneväni viikossa jo hyvin pitkälle. Aion kokeilla luokkien ja metodien toimintaa keskenään jo ennen algoritmin lisäämistä, tähänkin varmasti kuluu reippaasti tunteja. Seuraavaksi onkin vuorossa itse labyrintin luominen, johon oletan menevän ainakin pari viikkoa. Haluan edetä hitaasti ja välttää mahdolliset typerät virheet. Tallennusformaatin suunnittelun aloitan myös jo luokkien suunnittelun loppuvaiheessa, aika-arvion tekeminen jälleen hyvin vaikeaa, mutta varmasti menee ainakin muutama päivä. Loppupuolella aion toteuttaa ”forfeit”-toiminnon, joka kertoo käyttäjälle labyrintin ratkaisun, aikaa kuluu varmasti taas ainakin pari päivää.

7. Yksikkötestaussuunnitelma

Aion edetä pienin askelin, ja varmistaa että luokkatoteutukset toimivat keskenään. Ensimmäinen tavoitteeni on saada luokat rakennettua siihen vaiheeseen, että pystyn luomaan ”umpinaisen” labyrintin, jolloin voin myös testata hiiren sijoittamista ruutuun. Hiiren metodeja on myös luultavasti helppo testata jo melko alkuvaiheessa: sitä, ettei hiiri pysty menemään seinän läpi, että hiiri liikkuu oikein square-oliosta toiseen, ja että square-olio tallentaa hiiren sijainnin. Algoritmin ja graafisen toteutuksen suhteen koen olevani vielä melko pimennossa, mutta pyrin tekemään algoritmin suhteen vielä reippaasti taustatyötä, ja keskittyä implementoimaan sen huolella. Graafisen toteutuksen suhteen teen päätöksen vielä hieman myöhemmin.

8. Kirjallisuuslähteet ja linkit:

Weave-labyrintin rakenteesta: <https://weblog.jamisbuck.org/2011/3/4/maze-generation-weave-mazes.html>

Labyrinttien rakenteet ja algoritmit: <http://www.astrolog.org/labyrnth/algrithm.htm>

Infoa recursive backtracking -algoritmista: <https://brilliant.org/wiki/recursive-backtracking/>