

Yleiskuvaus

Olen tekemässä ohjelma, joka luo ja piirtää ruudulle labyrintin, sekä asettaa pelaajan labyrintin vapaaseen ruutuun. Käyttäjän tehtävä on ohjata hiiri pois labyrintistä. Hänellä on mahdollisuus luovuttaa, jolloin ohjelma näyttää oikea ratkaisu kuljettamalla hiiri pois labyrintistä. Labyrintti tukee myös toisiaan ylittäviä ja alittavia reittejä eli ei ole vain tasomainen. Labyrintin koon on vapaasti annettavissa. Myös labyrintti voidaan tallentaa ohjelmasta tiedostoon.

Käyttöohje

Ohjelma käynnistetään Eclipsen tai python konsolin avulla. Käyttäjä pystyy syöttämään nimen, muuttaa labyrintin kokoa ja lähtöpistettä. Myös on mahdollista katsoa, miten labyrintti luodaan ja asettaa "häntä", joka näyttää kuluneen matkan.

Pelin aikana käyttäjä näkee pelaajan ja konsolin, missä ovat pelin kommentit. Oikeassa alakulmassa ovat kaksi painiketta "New Game" ja "Give Up". Yllä on taulukko, jossa näkyy parhaat pisteet.

Ulkoiset kirjastot

Yritin käyttää vain sellaisia kirjastoja, jotka olemme käyneet kurssin aikana Labyrintin ja pelaajan piirtämisen käytiin QGraphicsRectItem sekä QGraphicsPolygonItem. Minulle uusi oli QMessageBox.

Ohjelman rakenne

Ohjelmalla on yhdeksän luokkaa. Main-luokassa luodaan labyrintti ja käynnistetään graafinen osuus. GUI-luokassa luodaan ikkuna ja asetetaan sekä ohjelman muuttumattomat asiat, kuten kello, konsoli, pisteytystaulukko ja kaksi painiketta että piirustus alue. Yksinkertaistamiseksi ja helpottamiseksi takia "start menu":lle on tehty oma luokka. Pelin aloittamissa "start menu" poistetaan ja asetetaan piirustus alue. Painamalla "New Game" -painiketta piirtoalue poistetaan ja luodaan aloitusvalikko. Pelaajalle sekä labyrintin neliölle on omat luokat, jossa luodaan niiden grafiikka. Myös score-luokka helpottamiseksi takia on tehty toiselle luokassa. Siellä pisteet luetaan tiedostosta sekä kirjoitetaan tiedostoon.

Algoritmit

Labyrintin sekä luomisen että ratkaisun on käytetty "recursive backtracker" algoritmi, jossa luodaan kaksi tyhjää lista. Molemmat tallentavat koordinaatit, joissa olemme jo olleet, mutta jos et voi enää siirtyä eteenpäin. Poistetaan yhdestä listasta viimeinen alkio ja yritetään taas liikkua, mutta otetaan huomion toisen listan eli paikan jossa olimme jo. Periaatteessa kuljetaan etempai ja palataan takaisin kunnes taas pystytään liikkua. Suunta valitaan satunnaisesti. Myös ohjelma laittaa loppu koordinaatin eli oveen pisin matkan lopussa. Sitä varten tarvitaan kolmas lista joka tallentaa pisin matkan. Neljäs lista tallentaa pelaajan matka. Sitä tarvitaan sen takia, että näytä pelaajan kulunut matka, kun on valittu "tail". Myös silloin kun lasketaan pisteet pelin lopussa.

Tiedostot

Ohjelmassa on käytössä tekstitiedostoja pisteytyksen ja labyrintin tallentamisen.

Testaus

Ohjelmalle en ole kirjoittanut erillinen luokka testaamisen varten. Testaukset on tapahtunut konsolin avulla kirjoittamalla ja poistamalla tietyt pikku koodit.

Ohjelman tunnetut puutteet ja viat

Ohjelma pystyy tallentamaan labyrintin tiedostoon, mutta ei voi sitä pirttaa uudestaan. Tiedoston lukemisessa ei ole ongelma vain labyrintin piirtämisessä. Se vaatii perus koodin muuttamista eli yhdistämällä kaksi osa koodia. Toinen ongelma on koskee labyrintin kokoa. Luomalla suuren labyrintin, sen lukeminen on vaikeutuu, koska neliöitä ovat liian pieniä. Sitä voidaan ratkaista niin, että luetaan vain tietty osa labyrintistä ja liikutaan pelaajan kanssa.

3 parasta ja 3 heikointa kohtaa

Yksi parasta kohtaa minun mielestä on se, että on mahdollistaa katsoa miten luodaan labyrintti. Toinen on aloitusvalikko joka vaihtuu piirustus alueen kanssa ja kolmas on labyrintin luomisen algoritmi.

Yksi heikoimmista kohdista on labyrintin koko, kun kartta on suuri. Toinen on labyrintin luominen tiedostosta. Kolmas haittapuoli voidaan katsoa ohjelman viestintä pelaajan kanssa konsolin avulla.

Poikkeamat suunnitelmasta

Aluksi haluaisin luoda labyrintti tiedostosta, jossa on jo tallennettu kartta. Mielestäni se oli liian helppo ja tylsä pelaaja näkökulmasta. Ei ollut paljon vaihtoehtoa. Myös toteutusjärjestys on alussa ollut heikko. Halusin ensin tehdä graafinen osuus ja siten aloittaa algoritmin sekä piirustus alueen luomisen. Se aiheutti sen, että tein kaksi kertaa gradisen osuuden, koska sitä oli vaikea yhdistää ohjelman algoritmin kanssa.

Toteutunut työjärjestys ja aikataulu

Ensimmäinen viikko perjantaina: luodaan ikkuna ja asetetaan napa painikkeita
Toinen viikko perjantaina: luodaan tarvittavat luokat ja jaetaan osittain tehtävät keskeen.
Kolmannella ja neljännellä viikolla en ole tehnyt mitään, koska oli työhaastatteluita ja tenttien valmistelu.
Viides viikko perjantaina sekä viikonloput: toteutin "recursive backtracker" algoritmin labyrintin luomisessa ja laitoin vähän kunnolle ulko näky.
Kuudennella viikolla aloitin keskiviikko illalla jatkoin torstaina koulun jälkeen ja siten koko perjantaina ja viikonloput: toteuttiin ohjelman itse ratkaisun, luoketelin ja yksinkertaisin jotkut osat, tein aloitusvalikko niin että se toimi piirustus alueen kanssa, kirjoitin koodi osa joka tallentaa labyrintin tiedoston ja lue sekä kirjoittaa score-tiedot, aloitusvalikon on siirretty toiselle luokalle helpottamiseksi takia.

Arvio lopputuloksesta

En ole tyytyväinen, koska pari asiaa jäivät tekemättä huonon aikataulun järjestämisen takia. Ohjelma olisi voinut tallentaa suuria kartoja, niin että myöhemmin niitä voisi jatkaa. Myös alku suunnittelussa oli "eväitä" joiden nostaminen pelin aika helpottaisi sen ratkaisua. Ohjelman rakenne soveltuu muutosten ja laajennusten tekemiseen, koska loppuviikosta yritin luokitella koodi niin että se kestäisi muutosta ja laajennusta.

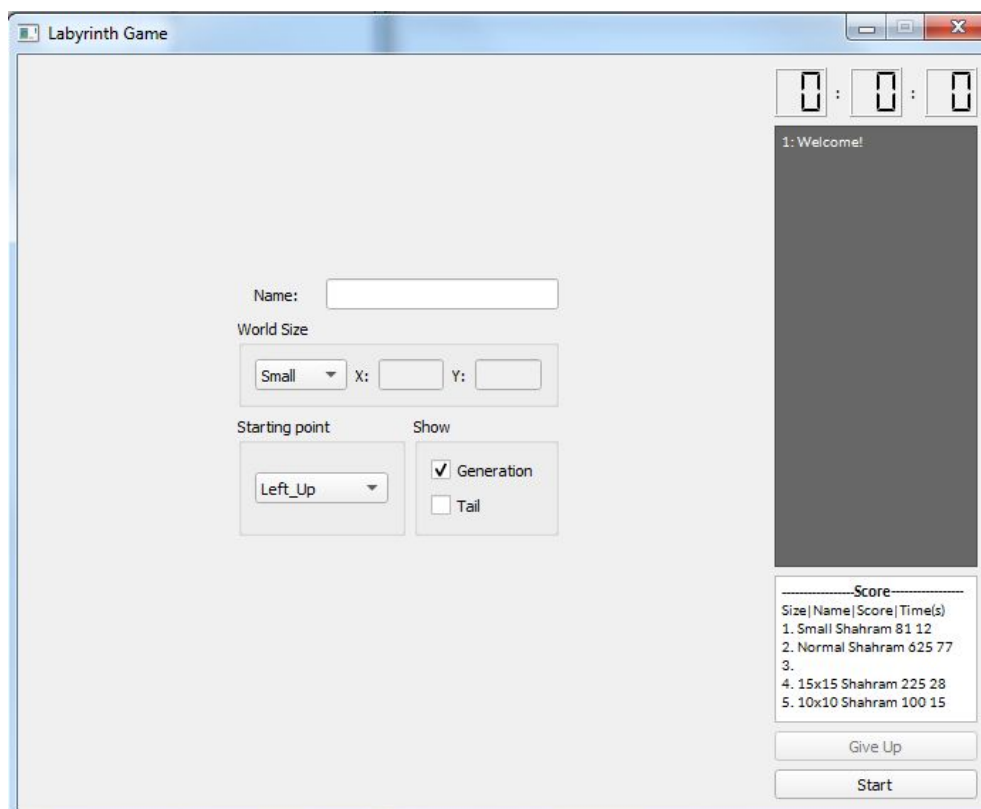
Viitteet

Suuri osa materiaalia, jotka olen käyttänyt, ovat perusopetuksen tarkoitettuja asioita. Myös suuri osa niistä ovat venäjän kielillä.

- Ohjelmoinnin perusteet Pythonilla, Teemu Sirkiä, 2015
- [TutorialsPoint-nettisivu](#) (Vain PyQt-QMessageBox)
- [Think Labyrinth](#) (Recursive Backtracker-algoritmi)
- [Crossplatform.ru](#) (luokkien ja niiden toimintojen ymmärtämiseksi)
- [Pythonworld.ru](#) (widget-esimerkkejä)
- [ZetCode.com](#) (widget-esimerkkejä mutta englanniksi ja vähän erilaisia)

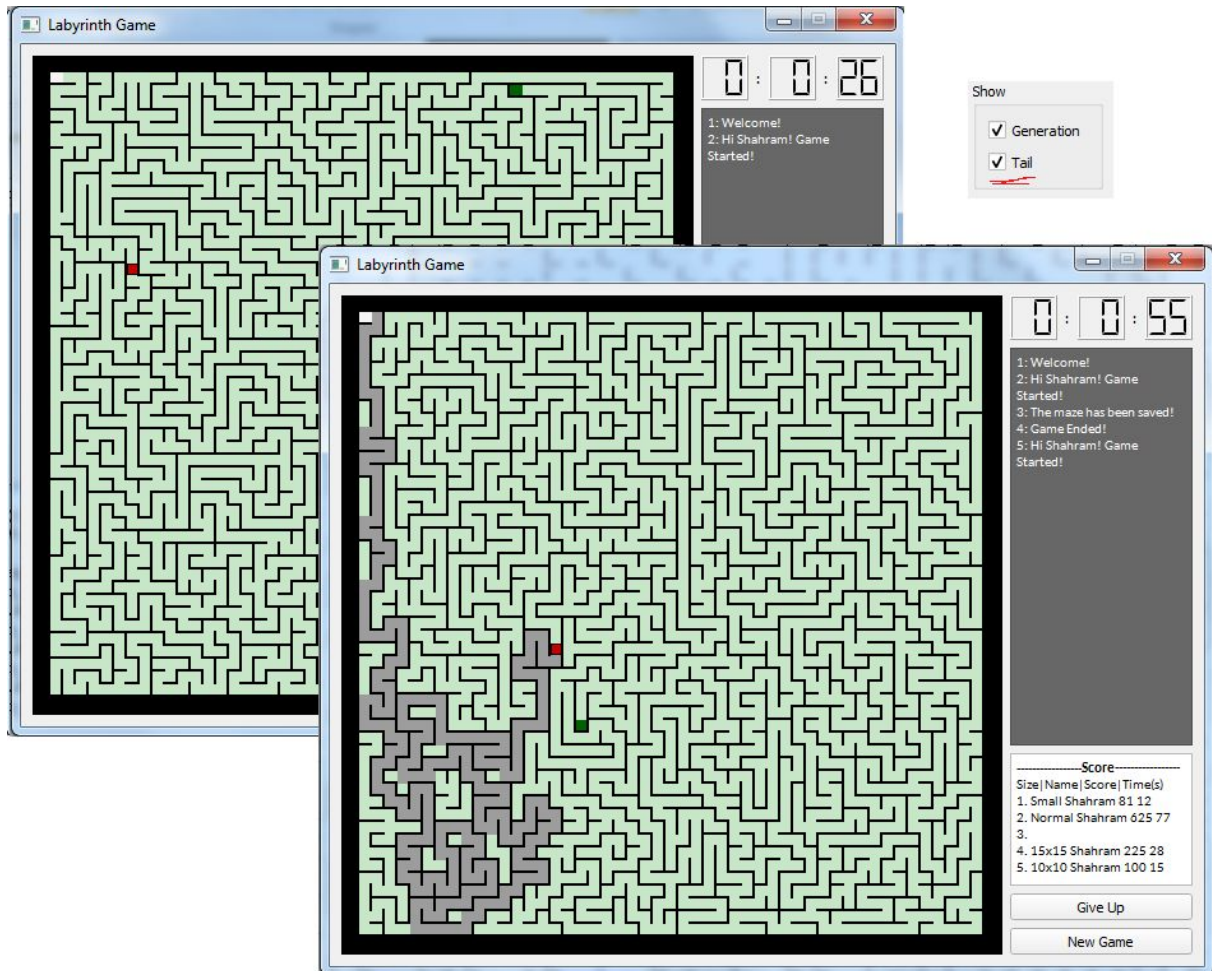
Myös halusin lukea [Making Games with Python&Pygame](#), mutta en ehtinyt.

Liitteet



```
def keyPressEvent(self, event):
    if self.isStarted and self.maze_is_ready == 1:
        key = event.key()
        if key == QtCore.Qt.Key_W or key == QtCore.Qt.Key_Up:
            self.player.try_to_move("UP")
        elif key == QtCore.Qt.Key_S or key == QtCore.Qt.Key_Down:
            self.player.try_to_move("DOWN")
        elif key == QtCore.Qt.Key_A or key == QtCore.Qt.Key_Left:
            self.player.try_to_move("LEFT")
        elif key == QtCore.Qt.Key_D or key == QtCore.Qt.Key_Right:
            self.player.try_to_move("RIGHT")
        self.checkPosition()
```





```

P main P gui P square P labyrinthworld maze.txt
1 Shahram:49:49:10:0:2:7
2 _s x|
3 x x|
4 x|
5 x|
6 x|
7 x|
8 _x x|
9 |
10 |x|
11 x x|
12 x x x|
13 x x x|
14 x x|
15 |x|
16 |x|
17 |x|
18 x x|
19 x x|
20 |x x|
21 x x|
22 x|
23 x x|
24 x|
25 x|
26 x| x x|
27 x x x| x|
28 |x x| x| x|
29 |x x| x| x|
30 |x x| x| x|
31 |x x| x| x|
32 x x| x x x| x|
33 x x x| x| x|
34 x x x| x x x| x|
35 |x x| x x| x x x| x|
36 |x x| x| x x| x x x|
37 |x x| x x| x x| x x| x|
38 |x x| x x| x x| x x|
39 |x x| x x| x x| x x|
40 |x x| x| x| x x| x x|
41 |x x| x| x| x x| x x|
42 |x x| x| x x| x x| x x|
43 |x x| x| x x| x x| x x|
44 |x x| x| x x| x x| x x|
45 |x x| x| x x| x x| x x|
46 |x x| x| x x| x x| x x|
47 |x x| x x| x x| x x|
48 |x x| x x| x x| x x|
49 |x x| x x| x x| x x|
50 |x x| x x| x x| x x|
51

```