

Numeerisen datan visualisointikirjasto

Tekninen suunnitelma – Ohjelmoinnin peruskurssi Y2

Akseli Oinaanoja 656551 (BIO 2.VSK)

6.3.2019

1. Ohjelma eroteltaisiin graafisen käyttöliittymäikkunan ja siihen liittyviin luokkiin. Luokkia voisivat olla esimerkiksi piirrettävät viivat, pisteet, itse kuvaikkuna ja muut visuaaliset aspektit kuten virheilmoitus- ja syöteikkunat. Suunnittelin ohjelman kokonaisuuteen mahdollisesti oman luokan `gui_reader`, joka lukisi tarvittavat tiedot käyttäjän nimeämästä tiedostosta. Tiedoston lukevassa luokassa voisi olla esimerkiksi metodeja, jotka hakevat raakadatan, tulostavat sen. Grafiikkakäyttöliittymän `graphics_scene` -luokka kokoaisi graafiset elementit, (pisteet viivat jne.) joita tarvitaan visualisoinnin luomiseen. Kuvaikkunan ja datan keskeisin elementti on pistettä kuvaava olio, jolla on kaksi attribuuttia, `x` ja `y`. `x` voi olla joko merkkijono, esimerkiksi "mansikka", ja `y` on aina numeerinen. Tämän `x:n` ominaisuuden avulla voidaan helposti luoda yhteyksiä asioiden ja numeroiden välille.
2. Käyttäjä haluaa luoda pylväsdiagrammin lähikaupan ruokien hinnoista. Hän luo `csv`-tiedoston, johon lisää tuotteen nimen ja hinnan jossakin tietyssä järjestyksessä, esimerkiksi juurikin ensin `x`, sitten `y` -järjestyksessä. Hän käynnistää ohjelman esimerkiksi komentoriviltä ja antaa syötteeksi luomansa tiedoston nimen. Tässä vaiheessa tiedoston lukeva `gui_reader` aktivoituu ja lukee tiedoston sisältämän avainsanan `#column` tai muun vastaavan. Syötteen lukeminen riippuu siis käyttäjän haluamasta visualisointitavasta. `Gui_reader` sisältää useamman eri tavan (metodin) lukea syöte (vaikkapa `read_column`, `read_line`, `read_pie`) ja niistä vain yhtä käytetään kerrallaan datan lukemiseen. Nämä tiedot siirretään `graphics_scenelle` ja tämä luo käyttäjän toiveesta riippuen kuvaikkunan ja pisteet. Tässä pylväsdiagrammin tapauksessa `x`-akselilla esiintyy tuotteiden nimiä, ja `y`-akselilla hinta. Jos käyttäjä haluaa lisätä selitteitä, muuttaa luettavaa tiedostoa tai vaikka tallentaa kuvaajan, se onnistuu helposti kuvaikkunan sisältämistä valintapalkeista. `Graphics_scene` sisältää siis vähintäänkin muutaman eri valintalaatikon, joista tapahtuu hieman eri asioita metodien avulla. `Graphics_scene` luokassa on metodeja, joilla saadaan tarvittaessa luettua kaikkien pisteiden `x`- ja `y`-arvot ja tallennettua ne listaan sekä alaluokka `graph_item`, joka sisältää yksittäisen pisteen tiedot ja se, mitä pisteen avulla tehdään (viiva, pylväs vai piirakka). Jokainen yksittäinen `graph_item` voitaisiin lisätä jonkinlaiseen yläluokkaan kuten `graph_array` ja tämä `graph_array` lopuksi tuodaan `graphics_scenelle` listaan.

3. Ohjelmassa hyödynnän csv-datan lukemisessa tietovirran erottelua pilkuin (`.strip(",")`) ja poimimista taulukkoon tai listaan, jonka koon voi helposti arvioida laskemalla pilkkujen määrän (tai tiedoston koosta). Ohjelma ei kuitenkaan käytä sen suurempaa laskentaa, eikä täten tarvitse juurikaan matemaattista optimointia tai laskentatekniikoiden miettimistä. Tiedoston sisältämän datankin lukeminen voidaan tehdä hyvin yksinkertaiseksi luomalla selkeät säännöt ja ohjenuorat käyttäjälle. Tällöin esimerkiksi ohjelman ei tarvitse jokaisen data-arvon kohdalla tarkistaa sen tyyppiä tai arvata, mihin toiseen arvoon se on liitetty (x,y). Mikäli data ei ole ohjeen mukaista, voidaan luoda vain error ja virheteksti käyttäjälle. Akselien sopivaan skaalaukseen tulee lukea data läpi ja selvittää maksimi- ja minimiarvot, jotta saadaan kaikki datapisteet sopivalle ja näkyville välille. Myöskin esimerkiksi pylväiden määrä ja niiden väliin jäävän tilan laskeminen tapahtuu ikkunan leveyden avulla. Tällaisissa tehtävissä aion kokeilla sopivaa kerrointa, joka näyttää visuaalisesti miellyttävältä ja käytän sopivaa.
4. Ohjelmassa käytetään pääasiassa täysin dynaamisia tietorakenteita. Ohjelma koostuu siis 4 tasosta, joista ylin on `gui_reader`in luoma `graphics_scene`. `Graphics_scene` sisältää `graph_arrayn`, joka puolestaan koostuu `graph_ite`meistä, jotka koostuvat pistekoordinaattiolioista ja niiden käyttötarkoituksesta. Näitä on kuvailtu ylempänä. Datapisteiden lukemisessa voidaan luoda valmiiksi taulukko, jonka kokoa laskiessa hyödynnetään pilkkujen määrää. Kuitenkin luettavaa tiedostoa voidaan muokata ja sinne voidaan lisätä dataa, jolloin tietorakenteiden koot voivat vaihdella paljonkin.
5. Uskoisin, että datan lukeminen tiedostosta on melko helppo toteuttaa. Ainakin tässä mainitut kolme eri metodia `gui_reader` luokkaan voisi olla melko nopeita. Tähän arvioisin kuluvan noin 5 tuntia. Vaikeimmaksi tulee mitä luultavimmin osoittautumaan `graphics_scenen` luominen sen sisältämine alaluokkineen ja visuaalisine painikkeineen, joille tulisi myös löytää oikeanlainen toiminnallisuus. Myöskin siis testaamiseen tulee kulumaan aikaa paljon. Ohjelman tulisi pystyä toteuttamaan ainakin osa mainitsemistani ominaisuuksista ilman ongelmia. Ominaisuuksien lopullinen määrä ja monimutkaisuus vaikuttavat työläysarvioon, mutta uskoisin, että nämä testaukset tulevat yhteensä viemään ainakin 20 tuntia. Itse olioiden, muuttujien ja metodien koodaaminen tulee viemään toiset 20 tuntia. Suunnittelin ohjelman rakentamisen suunnilleen edellä mainitussa järjesetyksessä eli ensin aloitan tiedon lukemisesta, etenen grafiikkaikkunan luomiseen ja olioiden ja metodien testailuun niiden luomisen ohella ja lopuksi lisäilen käyttäjän kannalta toivottavia ominaisuuksia, kuten zoomia, selitteitä, tallennusta ja niin edespäin.
6. Testatessa kiinnitän huomiota `gui_reader`in tiedostonlukumetodeihin: niiden tulee osata, päätellä virhe tiedoston datassa, näitä voivat olla esimerkiksi erikoismerkit. Päätellään tyhjän rivin olevan tiedoston loppu. Tiedoston lukemisen virheet on melko helppo kokeilla ilman, että muu ohjelma olisi vielä valmis. Muiden ohjelman osien testauksessa voisin hyödyntää tynkämetodeja, jotta suoritus ei pysähdy muualla virheeseen.

7. Olen suunnitellut käyttäväni ainakin PyQt:n dokumentaation apuna seuraavia nettisivuja: <http://pyqt.sourceforge.net/Docs/PyQt5/> ja <https://www.tutorialspoint.com/pyqt/>. Aion myös hyödyntää StackOverflown keskustelufoorumin apua.