

Random Walk -ohjelmointiprojekti syksy 2020

Aapo Hetemaa ja EV

Käyttöohje

Käyttöliittymä koostuu kuvaajasta, start/stop -nappulasta, nykyisen kuvaajan paikan osoittavista koordinaattikentistä, iteration times-laatikosta ja asetusvälilehdistä. Animaation voi käynnistää oletusasetuksilla start-nappulaa painamalla välittömästi ohjelman alettua, ja koko ajan koordinaattikenttiin päivittyy kuvaajan tämänhetkinen sijainti. Samaa nappulaa painamalla suoritus päättyy. Iteration Times -kentässä laskurin arvo kertoo kuinka monta kierrosta animaatiota on ajettu.

Animaation asetukset on jaoteltu välilehdille niiden tyyppin mukaan käyttöliittymän selkeyttämiseksi.

1) Ensimmäisellä välilehdellä on kuvaajan merkkiin liittyvät säädöt. Merkin muotoa, kokoa ja väriä on mahdollista muuttaa intuitiivisilla valintatoiminnoilla. Oletuksina on ympyrä, koko 5, väri punainen (keskeltä) ja musta (reunaviiva). Kaikkia näitä asetuksia voi muuttaa milloin tahansa, myös animaation ollessa käynnissä.

2) Toisella välilehdellä on animaation toiminnallisuuksiin liittyviä asetuksia. Step-kohdassa voi muokata satunnaiskävelyn askelen pituutta, jonka oletus on 0.1. Change in Axis -nuppia pyörittämällä voi säätää paljonko akseleita siirretään (välillä 0-10) aina, kun kävely hoippuroi kuvaajan rajojen ulkopuolelle. Tässä on oletuksena 0.5. Axis Limits -säätö ilmaisee, millä välillä kuvaajan akselit ovat animaation alkaessa (välillä 0.5-10.5). Säädössä on oikeastaan rajojen itseisarvot, eli esimerkiksi valittaessa 4.5, ovat sekä x- että y-akselien rajat alussa -4.5 – 4.5, sillä kävely alkaa aina origosta. Oletusrajoina on -0.5 – 0.5. Kaikkia näitä asetuksia voi säätää kävelyn aikana, mutta akselien alkurajat päivittyvät vasta seuraavaan suoritukseen. Akselinsiirron ja stepin muuttaminen vaikuttaa välittömästi.

3) Kolmannella välilehdellä voi valita kävelyn jättämän jäljen asetuksia. Käyttäjä voi valita, näytetäänkö ”askelet” eli edeltävät pisteet. Pisteitä yhdistävän viivan laatu, paksuus ja väri ovat muokattavissa. Jäljen näkymistä päälle tai pois ei voi vaihtaa kävelyn aikana, vaan ohjelma ilmoittaa käyttäjän väärästä syötteestä. Kaikki viivan laatuun liittyvät muokkaukset ovat ensimmäisen välilehden asetusten tyyliin muokattavissa milloin tahansa. Ainoa rajoitus on, ettei ole mahdollista valita sekä merkkiä että jälkiviivaa ’none’:ksi, vaan ohjelma ilmoittaa virheestä. Muutenhan mitään animaatiota ei näkyisi. Myöskään ei ole mahdollista valita jälkeä pois päältä ja merkkiä ’none’:ksi.

4) Neljännellä välilehdellä käyttäjällä on mahdollisuus muokata satunnaiskävelyn todennäköisyysjakaumaa. Satunnaiskävelyn kaava on $x = x_0 + \text{step} \cdot \cos(\text{theta})$, $y = y_0 + \text{step} \cdot \sin(\text{theta})$, missä kulma theta on satunnaisluku ja kerroin step edellä käsitelty ”askelen” pituus. Jakauman voi valita jatkuvaksi tai diskreetiksi. Jos jakauma on jatkuva, on kulma satunnaisreaaliluku väliltä $[0, 2 \cdot \pi)$. Diskreetin jakauman valinta antaa käyttäjälle mahdollisuuden valita alemmalla säätönupilla luvun n (väliltä 1-16), jonka mukaisesti täysi ympyrän kulma jaetaan n yhtä suureen osaan, ja nämä kulmat muodostavat tasajakautuneen diskreetin todennäköisyysjakauman mahdolliset arvot. Eli esimerkiksi valinta $n=2$ jakaa kulman 2π kahtia kulmiin 0 ja π , ja theta arvotaan aina niiden väliltä. Tämä valinta tuottaa diskreetin yksiulotteisen satunnaiskävelyn, jossa mahdollisia suuntia ovat vain oikea ja vasen. Valinta $n=4$ tuottaa neljä mahdollista kulmaa: 0, $\pi/2$, π ja $3\pi/2$ ja sen myötä kaksiulotteisen diskreetin satunnaiskävelyn, jossa seuraava askel on aina 0.1 mittainen (oletusasetuksella step = 0.1), ja mahdollisia suuntia on neljä: oikea, vasen, ylös, alas. Nupin oletusarvo onkin $n=4$. Jakauman tyyppiä ei voi vaihtaa jatkuvasta diskreetiksi tai toisinpäin ajon aikana, vaan ohjelma ilmoittaa virheestä. Diskreetin jakauman arvojen määrää voi silti säätää milloin tahansa, ja jos diskreetti suoritus on käynnissä, päivittyy uusi valinta välittömästi seuraaviin kävelyaskeliin.

Ohjelmoinnin ratkaisuja

Projektin tavoitteena on täyttää arvosanan 4 vaatimukset. Iteraatiokertojen määrän laskeminen on ylimääräinen ominaisuus, ja muuten vaaditut asiat on tehty.

Itse plotti luodaan erillisessä funktiossa createPlot tai createAnimation -funktioilla sen mukaan, onko käyttäjä valinnut vanhat pisteet eli trailin jätettäväksi näkyviin, jolloin luodaan animatedline. Myös randomluku theta lasketaan omassa funktiossaan countRandomAngle. Oma funktionsa on virheilmoitusdialogin luova printErrorDialog, jota hyödynnetään eri tilanteissa, kun käyttäjän "syöte" on virheellinen. Funktio ottaa parametrinaan minkä tahansa tilanteeseen sopivan virheilmoitustekstin.

Plotti/animaatio säilytetään luokan private propertyinä samoin, kuin jotkin muut keskeiset muuttujat, joita halutaan käsitellä missä tahansa luokan metodeissa ja widgettien callbackfunktioissa. Tärkein ja pisin näistä callbackeista on startButtonPushed-metodi, jota kutsutaan, kun start-nappulaa on painettu, ja jolla luodaan plotti/animaatio.