



Processing – lisätekemistä kiinnostuneille

Olemme aiemmissa ohjeissa arponeet sanoja valmiista listoista ja sijoitelleet niitä toistensa perään. Tämä onkin luonteva ensimmäinen askel tietokoneavusteisessa runoilussa. Tässä ohjeessa on hieman lisäideoita tekemiseen.

1. Tekstiaineistojen lisääminen

Sanojen keksiminen ja lisääminen listoihin yksitellen on hieman uuvuttavaa. Mitä jos voisit ladata mukaan valmiin aineiston ja arpoa sanoja siitä? Aineisto voisi olla esimerkiksi Kalevala, Shakespearen näytelmä tai vaikka oma pro gradusi.

Tämä on täysin mahdollista. [Tutustu esimerkkikoodiin](#). Tutustu etenkin koodin kommentteihin, joiden kautta opit ymmärtämään ohjelmaa ja myös muokkaamaan sitä.

Käytännössä ei ole järkevää arpoa täysin satunnaisia sanoja esimerkiksi kirjasta, sillä tällöin lauseeseen voi tulla vaikkapa viisi verbiä peräjälkeen. Voit kuitenkin arpoa aineistosta kokonaisia lauseita. Toinen vaihtoehto on ladata mukaan useita aineistoja. Ensimmäisessä aineistossa voi olla pelkkiä substantiiveja, seuraavassa pelkkiä verbejä jne.

2. Runoilu ja tekoäly

Tässä projektissa olet arponut sanoja listoista ja asettanut niitä peräkkäin. Edellisen kohdan ohjeilla voit muodostaa listoja jonkin valmiin tekstiaineiston sanoista. Vaikka tekisitkin tämän, on edessäsi kuitenkin seuraavanlaisia haasteita.

- Miten tietokone voisi tunnistaa, onko sana substantiivi, verbi, pronomini vai adjektiivi?
- Miten tietokone voisi tunnistaa, miten sanat lausutaan ja mitkä sanat rimmaavat keskenään?
- Voisiko tietokone osata tulkita sanojen välittämää tunnetta ja esimerkiksi vaihtaa tekstin väriä tunnelman mukaan?
- Voisiko tietokone tarvittaessa myös taivuttaa sanoja?

Yllä listatut asiat ovat ohjelmoinnin näkökulmasta jo huomattavasti mutkikkaampia kuin tämän projektin aikana ratkotut haasteet. Haasteita ei oikeastaan kannata lähteä ratkomaan itse tyhjästä alkaen. Huomattavasti järkevämpää on valita jokin valmis kirjasto, jossa on tarvittavia toimintoja. Kirjastolla tarkoitetaan ohjelmoinnissa joukkoa komentoja, jotka joku toinen on tehnyt valmiiksi.

RiTta-kirjasto

Eräs hyvä kirjasto on *RiTta* – *A software toolkit for computational literature* ([linkki RiTtan esittelysivustolle](#)).

RiTtassa on oma kirjasto JavaScriptille (*rita.js*), mutta myös Javalle. Koska Processing on Java-pohjainen kieli, tulee käyttää Java-kirjastoa ([latauslinkki Java-kirjastoon](#)).

Seuraavaksi sinun tulisi lisätä kirjasto eli *rita.jar*-tiedosto oikeaan paikkaan. Processingin kirjastot ovat samassa Sketchbookissa, johon työtkin tallentuvat. Niille on siellä oma *libraries*-kansionsa. Tarkista Sketchbookin sijainti valitsemalla **File > Preferences > Sketchbook location**.

Siirrä *rita.jar*-tiedosto *libraries*-kansioon. Luo *libraries*-kansioon *rita*-kansio ja sen sisälle vielä *library*-kansio. Tiedostopolun pitäisi siis olla seuraavanlainen: **Sketchbook/libraries/rita/library/rita.jar**.

Tämän jälkeen voit valita Processingissa **Sketch > Import Library**. *rita*-kirjaston pitäisi löytyä listasta. Klikkaa sitä.

Tutustu sitten [Processing-esimerkkeihin RiTtan Github-sivustolla](#). Jokainen kansio on oma esimerkinsä. Kansio sisältää *.pde*-päätteisen Processing-ohjelman, jonka koodin voi sellaisenaan kopioida Processingin editoriin. Tämän lisäksi kansio saattaa sisältää *data*-kansion, jossa on esimerkiksi käytettäviä tekstiaineistoja.

Monet esimerkit hyödyntävät valmiita tekstiaineistoja, kuten hupaisa Kafkan ja Wittgensteinin tuotantoa yhdistelevä Kafgenstein. Jos haluat kokeilla tätä esimerkkiä, sinun on varsinaisen koodin lisäksi kopioitava tekstiaineistot *data*-kansioon, aivan kuten teit mukaan tuomiesi kuvan ja fontinkin kanssa.

Osa esimerkeistä ei kuitenkaan käytä tekstiaineistoja. Esimerkiksi rimmaavia sanoja etsivä *RandomRhymes* on katsomisen arvoinen!

Tekijä:

Eloholma, M. (2019). *Processing – lisätekemistä kiinnostuneille*. LUKILOKI-hanke 2018–2022. Jyväskylän yliopisto.

Sisällön on muokannut tämän dokumentin muotoon LUKILOKI-tiimi.