Tiralabran viikkoraportti #2 23.5.2013 Mikko Kangasmäki
Kuvan pakkaus

Tällä viikolla jatkoin kokeilujani kuvankäsittelyssä. Löysin bugin, jonka ansiosta enintään yhdessä pikselissä esiintyvän värisävyn poistaminen kuvasta oli toiminut katastrofaalisen huonosti. Korjattuani sen vertailin silmämääräisesti erilaisia tapoja vähentää esiintyvien värisävyjen määrää. Kaikki kokeilemani tavat olivat melko yksinkertaisia.

Laskeskelin myös testikuvassa esiintyvien saman sävyn jonojen määriä, ja miten pikseleiden uudelleensävyttäminen vaikuttaa siihen.

M.H:n kehotuksesta lataan kokeilutiedostotkin GitHubiin, vaikka ne ovat jokseenkin lukukelvottomia.

Kuvankäsittelyn lisäksi aloitin kokeilut aallokkeiden kanssa. Tosin, siinä toistaiseksi olen laskeskellut vain eri välien summia. Ajattelin, että en välitä vielä aallokkeiden laskemisesta, vaan enemmänkin niiden tallennusmuodosta. Ne voi tallentaa kaksiulotteiseen ja selkeään taulukkoon, joka vie paljon tilaa, tai yksiulotteiseen pieneen taulukkoon, jonka kanssa joutuu pelleilemään indekseillä. Tilan säästössä jälkimmäinen on tietenkin toivottavampi.

Sitten aloin kirjoittaa lopullista koodia: Kirjoitin olion, joka lukee kuvan byte-taulukoksi, ja joka pystyy myös tallentamaan sen. Tässä tuli yllättäviä ongelmia, sillä jostain käsittämättömästä syystä se, mikä toimi kokeilukoodissani ei enää toiminut tässä. Periaattessa ongelma oli siinä, että int-arvo 200 esimerkiksi muuttuu bytessä – 56;ksi, ja kun se muutetaan takaisin intiksi, arvo pysyy -56:na.

Ongelman ratkaisuhan ei ole erityisen vaikea, mutta olisin mieluummin tehnyt sen räpellyksissäni toimineella mentelmällä kuin if-lauseella. Tämä laittoi minut myös harkitsemaan shorttien käyttöä bytejen asemesta, koska tämä byten ominaisuus on omiaan aiheuttamaan myöhemmin vaikeita ongelmia. Toisaalta, shorttien käyttö veisi ~3500x2000 pikselin esimerkkikuvassanikin seitsemän megaa enemmän muistia.

Valmista koodia ei tullut tosiaankaan paljoa, ja vielä pahemmaksi tilanne muuttui, kun aloin kirjoittaa testejä. Huomasin, että suunnitelmani ohjelman rakenteelle oli kehno. Ne metodit, joita olisi ollut eniten syytä testata olivat privaatteja. Kun yritin selvittää, kuinka sellaisia testataan JUnitilla, törmäsin ajatukseen, että tämä on “code smell”. Ja tosiaankin, kaikki ne metodit olivat sellaisia, jotka voisivat olla omassa luokassaan, koska niille todennäköisesti on käyttöä muuallakin.

Aloin siis uudelleenjärjestellä ohjelmaa, mikä puuha on vielä kesken, enkä osaa vielä tarkkaan sanoa, miltä tänään jättämäni palautus tulee näyttämään.

Ajattelin, että minulla olisi jokaiselle tiivistämismuodolle, oma luokkansa, joka huolehtii datan tiivistämisestä, tallentamisesta ja aukaisemisesta. Vastaavasti bmp:lle pitää olla oma. Jokaisen luokan käytössä olisi apuvälineitä sisältävä luokka.

Kirjoitan koodin ja javadocin englanniksi silläkin uhalla, että se heikentää laatua, koska englannin käyttö on kuitenkin alalla standardi, ja siihen on varmaankin hyvä totutella ajoissa.