Pienoisrautatie, Tekninen suunnitelma

02.03.2015
Ossi Galkin 347637, AUT, 2012

Ohjelman rakennesuunnitelma

Ohjelma koostuu seuraavista osakokonaisuuksista:

Core_Systems

Luokka huolehtii kommunikoinnista käyttöjärjestelmän kanssa, sekä luo tarvittavat muut oliot ohjelman käynnistyessä. Ohjelman käynnistyessä se luo käyttöliittymän, tyhjän kartan, sekä Alustaja luokan avustuksella lataa ohjelmaan tulevat kuvaukset ratapalasista.

Tarjoaa metodit:

Aloita ohjelma:	suorittaa käynnistysrutiinit
Aja:	pyörittää ojelmaa

Käyttöliittymä

Huolehtii asioiden piirtämisestä, sekä hiiren ja ehkä myös näppäimistö käsittelystä. Tämän moduulin toteutus on vielä itselleni hieman epäselvää, sillä en ole perehtynyt vielä QT-ohjelmointiin ja siksi en lähde arvaaman siihen kuuluvia metodeja, muuten kuin että se sisältää metodit jossa näytölle piirrettäviä kohteita voi lisätä, poistaa ja päivittä, sekä niihin voidaan liittää toiminnalisuus tai tieto, kuten onko kohde ratapalikka tai "Tallenna" nappi ja mitä pitää tehdä kun kohdetta painaa. Lisäksi jokin metodi päivittää näytön. Käyttöliittymä saatetaan myös jakaa useampiin moduuleihin.

Nappulat

Luokka asioista jota käyttäjä voi klikata ruudulla. Nappula voi sisältää teksitä, kuten "Tallenna".

Tarjoaa metodit:

Luo:	luo uuden nappulan.
Klikkaus:	reagoi käyttäjän hiirenpainallukseen esimerkiksi tallentamalla radan.

Alustaja

Luokan metodit lataavat tiedot ohjelmaan tulevista ratapalikoista ohjelman mukana olevasta xml tiedostosta, sekä luovat niiden perusteella ohjelman ratapalikka oliot.

Tarjoaa metodit:

Alusta ohjelma:	lataa ratapalsten tiedot xml tiedostosta, sekä lisää ne kartalle.

Kartta

Kartta sisältää tiedon kaikista lisätyistä ratapalasista.

Tarjoaa metodit:

Lisää palanen:	Lisää ratapalan kartalle.
Poista palanen:	poistaa ratapalan kartalta.
Hae palanen:	palauttaa sijainnissa olevan plasen.

Ratapalikka

Geneerinen ratapalikka luokka josta luotuja olioita voidaan sijoittaa kartalle ja liittää toisiinsa. Ratapalaset sisältävät tiedon mille ne näyttävät.

Tarjoaa metodit:
Luo: luo ratapalasen.
Poista: poistaa ratapalasen.
Päivitä: päivittää ratapalikan sijainni tai asennon.
Lisää kontaki: Liittää ratapalikan toiseen.

Tallenna_Rata

Tallentaa kartalla olevat ratapalaset xml tiedotoon.

Tarjoaa metodit:
Tallenna: tallentaa radan.

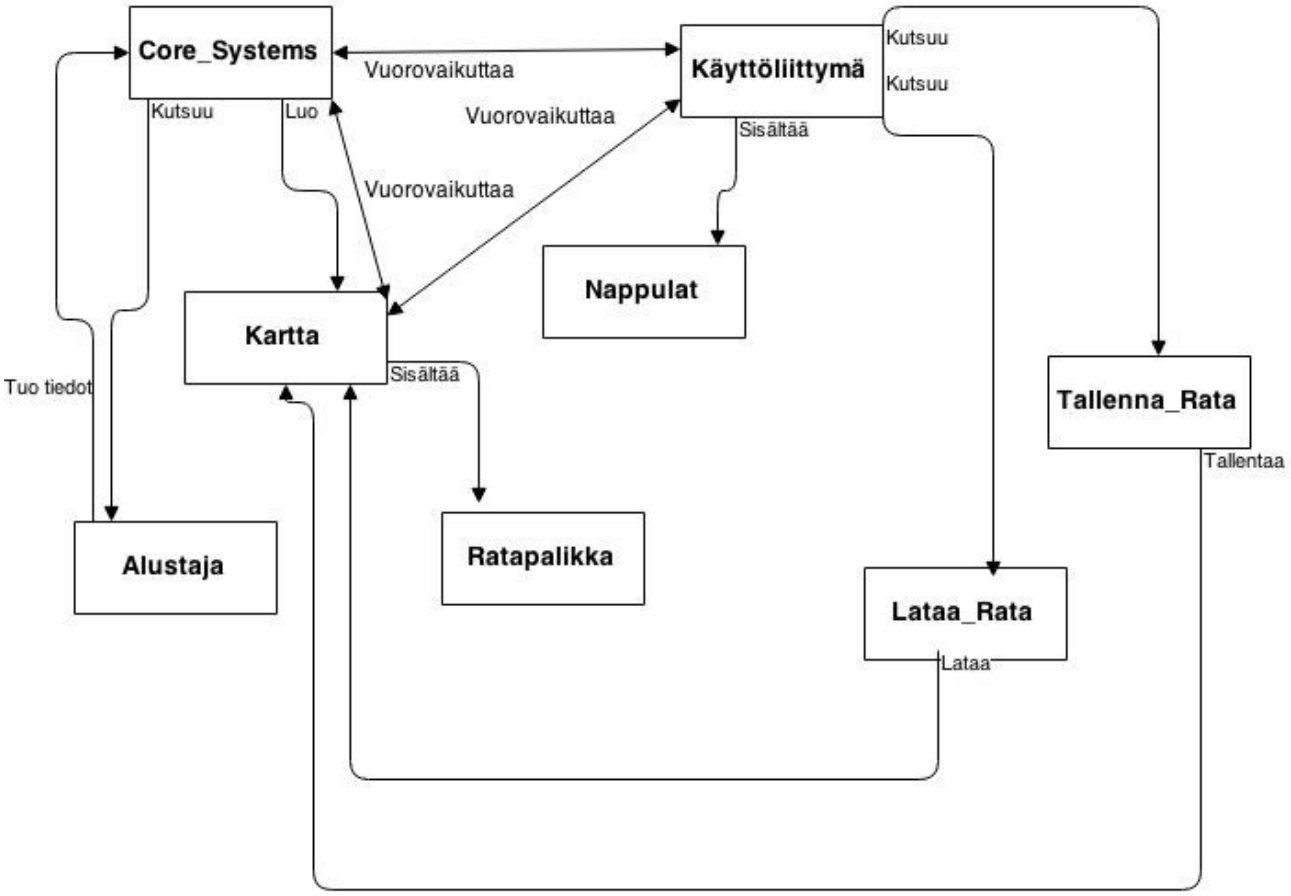
Lataa_Rata

Lataa tallennetun radan ja lisää ratapalaset kartalle.

Tarjoaa metodit:
Lataa: lataa radan.

Juna
Mahdollisesti toteutettava luokka joka luo junan joka osaa ajaa rataa pitkin eteenpäin. Luokka toteutetaan jos jää aikaa ja siksi sen tarjoamia moduuleja ei ole listattu.

Yksinkertaistettu kuva luokkien suhteista:



Käyttötapauskuvaus

Aivan aluksi avataan ohjelma ja sen jälkeen ladataan aiemmin tehty rata, jonka työstöä jatketaan. Tämän jälkeen rataan tehdään muutoksia ja yksinkertaisuuden vuoksi siitä poistetaan nyt aluksi vain yksi ratapalanen ja sen jälkeen rataan lisätään uusi pala joka muodostaa silmukan. Nämä asiat tehtyään käyttäjä tallentaa radan ja sulkee sitten ohjelman.

Ohjelma alkaa luomalla Core_Systems olio. Tämän jälkeen se luo käyttöliittymän, tyhjän kartan sekä kutsuus Alustaja luokan oliota lataamaan ohjelmassa olevat ratapalaset. Tämän jälkeen kutsutaan Core_Systems olion metodia joka käynnistää ohjelman varsinaisen "pääloopin", jossa eri päivitys rutiineja kutsutaan oikeassa järjestyksessä. Kun käyttäjä painaa "Lataa" nappulaa, käyttöliittymä huomaa sen ja kutsuu nappulan Klikkaus metodia, joka luo ja ajaa Lataa_Rata olion. Kaikki muut eri nappuloiden painamiset tapahtuvat tällä periaatteella ja siksi niitä ei kuvata enää jatkossa. Painalluksen seurauksena kartalle ladataan aiemmin luotu rata käyttäjän valitsemasta tiedostosta. Kun käyttäjä klikkaa kartalla olevaa palasta käyttöliittymä hakee oikean ratapalan käyttäen kartan hae pala metodia. Tällöin käyttäjän toiminnot kohdistuvat valittuun palaan. Kun käyttäjä painaa "Poista" nappulaa valittu ratapala poistetaan kartalta. Painamalla valikossa olevia ratapaloja käyttäjävoi valita ko. ratapalan lisättäväksi kartalle ja voi vetää sen haluamaansa kohtaan kartalla. Ratapala voidaan yhdistää toisiin paloihin viemällä se riittävän lähelle niitä, jolloin. Käyttöliittymä huolehtii tästä ratapalasten yhdistelystä. Kun ratapalat on yhdistetty toisiinsa käyttäjä painaa "Tallenna" nappia, jolloin Tallenna_Rata olio tallentaa kaikki kartalla olevat palaset xml tiedostoon. Lopeta napin painaminen aiheuttaa ohjelman lopettamisen.

Algoritmit ja Tietorakenteet

Ohjelma on laskennallisessa mielessä hyvin yksinkertainen se osaa piirtää asioita näytölle ja liittää ratapalikoita toistensa perään, sekä ladata ja tallentaa tuotokset. Se ei siis sisällä mitään laskennallisesti vaativaa kuten tekoälyä, kolmiulotteista maailmaa, törmäystarkistusta tai muuta vastaavaa ja tämän vuoksi ohjelmani käyttämät algoritmit ja niihin liittyvät tietorakenteet ovat hyvin yksinkertaisa, jopa niin yksinkertaisia että algoritmeista puhuminen vaikuttaa hieman yliampuvalle.

Kartalle lisättävien ratapalikoiden määrä on tyypillisesti joitakin kymmeniä ja suurimmillaa satakunta kappaletta. Tämän vuoksi kaikki kartalle lisätyt ratapalaset voidaan tallentaa listaan josta ratapalaset voidaan käydä läpi lineaarisesti, silloin kun testataan yhdistyykö lisättävä ratapalanen jo olemassa olevaan rataan. Jos rata muodostaa silmukan niin silloin kaikki silmukkaan kuuluvat palat on yhdistetty molemmista päistä, tämän huomaaminen ei vaadi erikoisempaa algoritmia. Kaikkien ohjelamani objektien määrä jota joudutaan piirtämään ruudulle on suurimmilla satakunta kappaletta ja tämän vuoksi kaikki objektit voidaan siirtää näyttöpuskurin välittämättä siitä näkyvätkö ne näytöllä vai eivät. Näytölle piirrettäviä objekteja ei siis karsita mitenkään. Ohjelma lataa kaikkien ratapalasten tiedot ja kuvaukset xml tiedostosta, sekä se voi ladata ja tallentaa tehdyn radan xml tiedostoksi.

Ratapalanen on keskeisessä roolissa ohjelman toiminna kannalta ja siksi se kuvataan tarkemmin. Ratapalanen sisältää tiedon sijainnistaan ja kulmastaan, siitä mille se näyttää, montako liitosta sillä voi olla, sekä mihin muihin palasiin se on liitetty. Lisäksi ratapalaseen saatetaan lisätä tieto apupisteistä joita pitkin mahdollisesti toteutettava juna ajaa rataa pitkin.

Aikataulu

Esiksi toteutetaan luokat Core_Systems, Ratapalikka ja Kartta. Nämä luokat sisältävät ohjelman ydin toiminnallisuuden ja siksi ne täytyy tehdä ensin. Näiden toteuttaminen ja testaaminen vie arviolta 40 tuntia.

Seuraavaksi tehdään luokat Käyttöliittymä ja Nappulat. Nämä luokat muodostavat käyttöliittymän ja siksi ne on tehtävä yhtäaikaa ja ydintoiminnallisuuksien jälkeen. Näiden toteuttaminen ja testaaminen vie arviolta 25 tuntia.

Viimeiseksi luokat Alustaja, Lataa_rata ja Tallenna_Rata luokat. Kaikki nämä luokat liittyvät tiedon tallentamiseen ja lataamiseen levylle, joten muut ohjelman osat eivät riipu niiden toiminnasta ja siksi ne voidaan tehdä viimeisenä. Näiden toteuttaminen ja testaaminen vie arviolta 15 tuntia.

Yksikkötestaussuunnitelma

Jokainen ohjelman metodi voidaan testata ykinkertaisilla yksikkätesteillä: antamalla niille yksinkertainen testi syöte ja tarkastelemalla miten olion tila tai funktion paluuarvo riippuu siitä. Tälläisiä testejä ovat esimerkiksi: Ratapalikka luokan luo metodille testi jossa tarkastellaan onko luotu palikka halutun lainen tai päivittääkö saman luokan päivitä metodi ratapalikkaa oikein. Kartta luokkaa voidaan testata lisäämällä kartalle ratapalasia ja kokeilemalla löytääkö haku sitten niitä. Kun palaset sitten poistetaan ei haun enää pitäisi niitä löytää.

Kirjallisuusviitteet ja linkit

ElementTree luokan käyttö
https://docs.python.org/3.4/library/xml.etree.elementtree.html

XTrackCAD ohjelma on jonkin lainen ohjelmani esikuva, se on innoittanut monia ohjelmani ominaisuuksita
http://www.xtrkcad.org/Wikka/HomePage
http://www.n-club-finland.org/wiki/Default.aspx?Page=xtrkcad