**IndieXML**

**Käyttöliittymä Harjoitustyö**

Rami Ojala

Harjoitustyö

Huhtikuu 2017

Tekniikan ja liikenteen ala

Insinööri (AMK), Tieto ja viestintätekniikka

Ohjelmistosuunnittelu

**Sisältö**

[1 Johdatus 3](#_Toc480921353)

[2 Taustaa 3](#_Toc480921354)

[3 Alustavat suunnitelmat 4](#_Toc480921355)

[4 Toteutus vaiheet ja ongelmat 6](#_Toc480921356)

[4.1 Modulaarisuus 7](#_Toc480921357)

[4.2 XML Import ja Export 7](#_Toc480921358)

[4.2.1 Oma koodi 7](#_Toc480921359)

[4.3 Binääri tiedostot 9](#_Toc480921360)

[4.4 Perus toiminnot 9](#_Toc480921361)

[4.5 Datagrid contextmenu 10](#_Toc480921362)

[5 Tilanne 10](#_Toc480921363)

[6 Pohdinta 11](#_Toc480921364)

[7 Itsearvio 11](#_Toc480921365)

[Lähteet 12](#_Toc480921366)

**Kuviot**

[Kuvio 1 Alustava luuranko 4](#_Toc480921346)

[Kuvio 2 IndieXML:n esi-isä, Windows Forms DBManager 4](#_Toc480921347)

[Kuvio 3 Pääohjelma ja plugin suunnitelma 5](#_Toc480921348)

[Kuvio 4 Jokin satunnainen kohta projekti taulusta. 6](#_Toc480921349)

[Kuvio 5 Lopullinen UML Kaavio suunnittelun loppuvaiheilla 6](#_Toc480921350)

[Kuvio 6 Malli XML tietokanta 8](#_Toc480921351)

[Kuvio 7 Contextmenu wireframe 10](#_Toc480921352)

# Johdatus

IndieXML on suunniteltu pienille pelinkehitys yrityksille, jotka tarvitsevat jonkinlaista tietokantaa peliin, johon SQL-tietokanta on ylilyönti

IndieXML on tarkoitettu yleistämään XML:n käyttöä pieniin peleihin. Sen tarkoituksena on helpottaa kehittäjiä luomaan ja ylläpitämään XML tietokantaa yksinkertaisessa Datagrid näkymässä.

IndieXML on vanha Windows Forms projekti, joka tehtiin yksinkertaisen pelin tietokannan luontiin. Tämä vanha projekti on nyt otettu uudistettavaksi perusteista alkaen Käyttöliittymä ohjelmointi kurssille, harjoitustyönä. Jos ohjelma joskus julkaisuun päästyään, löytää käyttäjänsä niin olen myös avoin muuttamaan ohjelman muihin käyttöjärjestelmiin.

# Taustaa

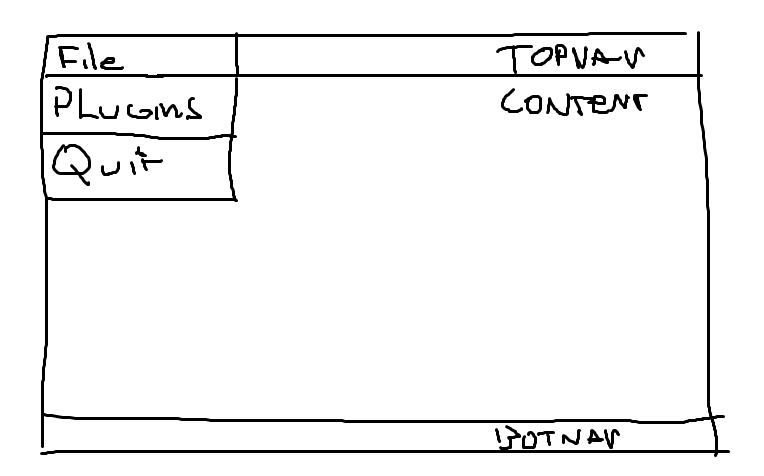
Idea tuli joskus viimekesänä jo itseasiassa mieleen, kun rupesin suunnittelemaan omaa peliä. Tiesin että jonkinlaista ns. tietokantaa tulee tarvitsemaan ja tiesin että SQL tietokanta on ylilyönti. Muistelin erään pelin käyttävän XML tyylistä tiedontallennusta (Freelancer) joten ajattelin hyödyntää XML tyylistä tietokantaa.

XML editointi sovelluksia on joka lähtöön. Tietokanta tyyppistä ratkaisua en itse löytänyt juuri sillä hetkellä, joten aloin sitä itse toteuttamaan. Ensimmäinen versio oli Windows Form tyyppisenä sovelluksena, vuonna 2016 kesällä, joka jäi hyllylle heinäkuun korvilla.

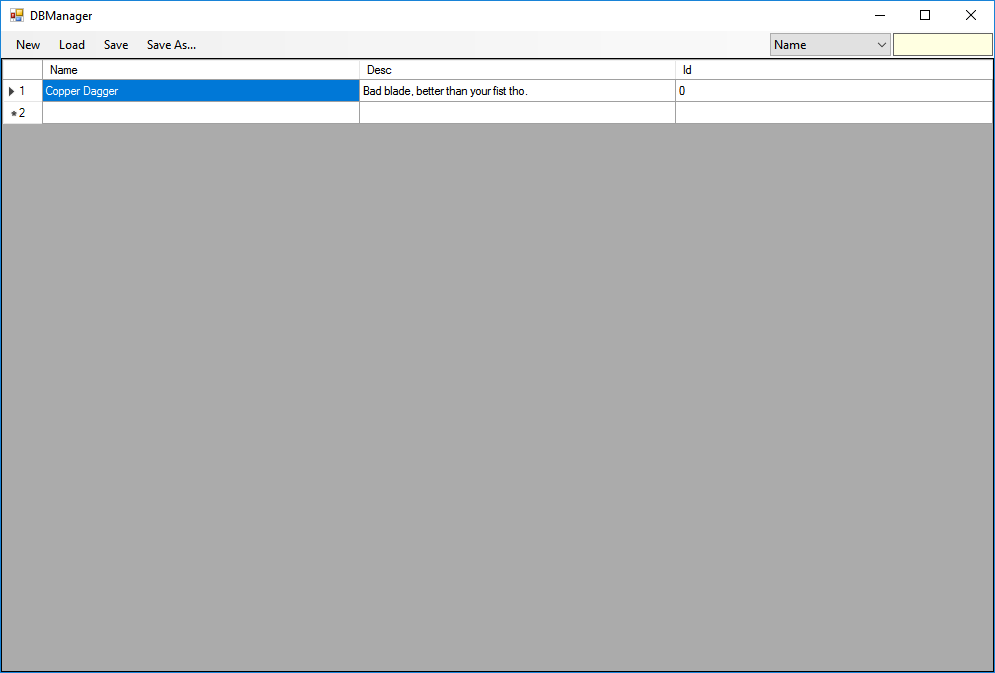
XML database editori palautui mieleen, kun mietin mitä harjoitustyötä alan tekemään. Päätin toteuttaa XML editorin nyt harjoitustyön ohessa kunnolla. Nimesin projektin IndieXML, koska ajattelin, että tällaista ohjelmaa voisi käyttää Indie pelikehityksessä pieniin tietokanta tarpeisiin.

# Alustavat suunnitelmat

Alustava suunnitelma oli, että ohjelmalla voisi tehdä yhden ”käsityksen” tietokantaan. Ohjelma sisältäisi reaaliaikaisen haun ja tietokantaan voisi lisätä ja poistaa kolumneja.



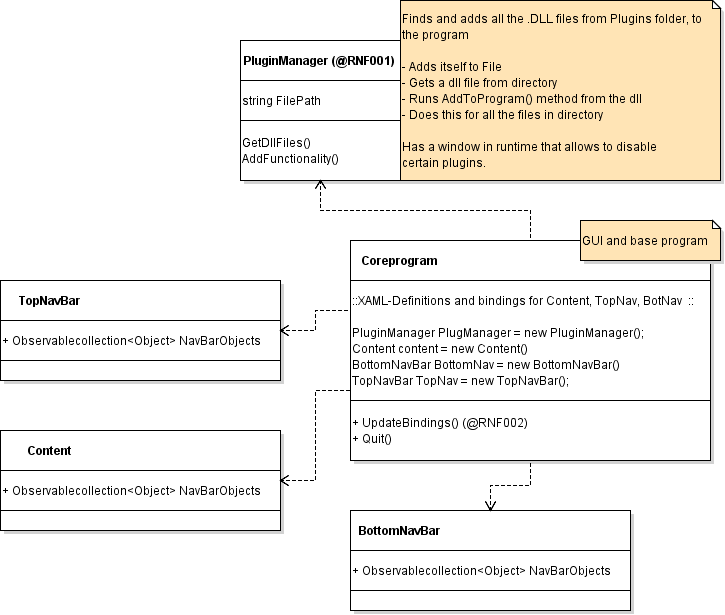
Kuvio 1 Alustava luuranko



Kuvio 2 IndieXML:n esi-isä, Windows Forms DBManager

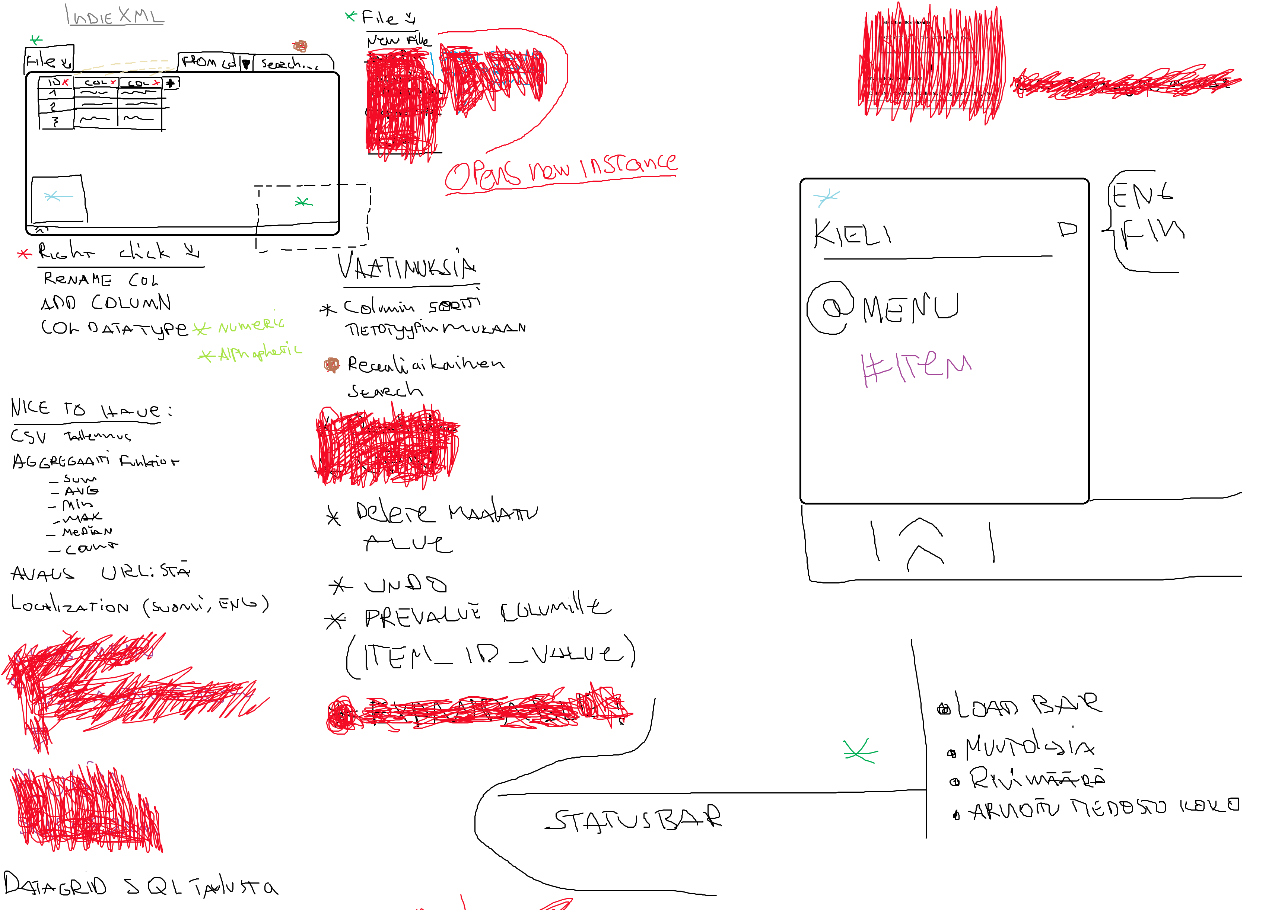
Uutta versiota lähtiessäni suunnittelemaan päätin, että laajennettavuus olisi tärkeä osa ohjelmaa. Kehittäjät voisivat itse lisätä toimintoja, jos näkevät jonkun asian tarpeelliseski.

Suunnittelin WPF-ohjelman niin, että siihen voi syöttää käyttöliittymään toiminnallisuutta, kirjoittamalla omia DLL tiedostoja. Suunnittelin myös Interfacen jonka jokainen luokka pitää toteuttaa, että ohjelma lataa sen sisään.

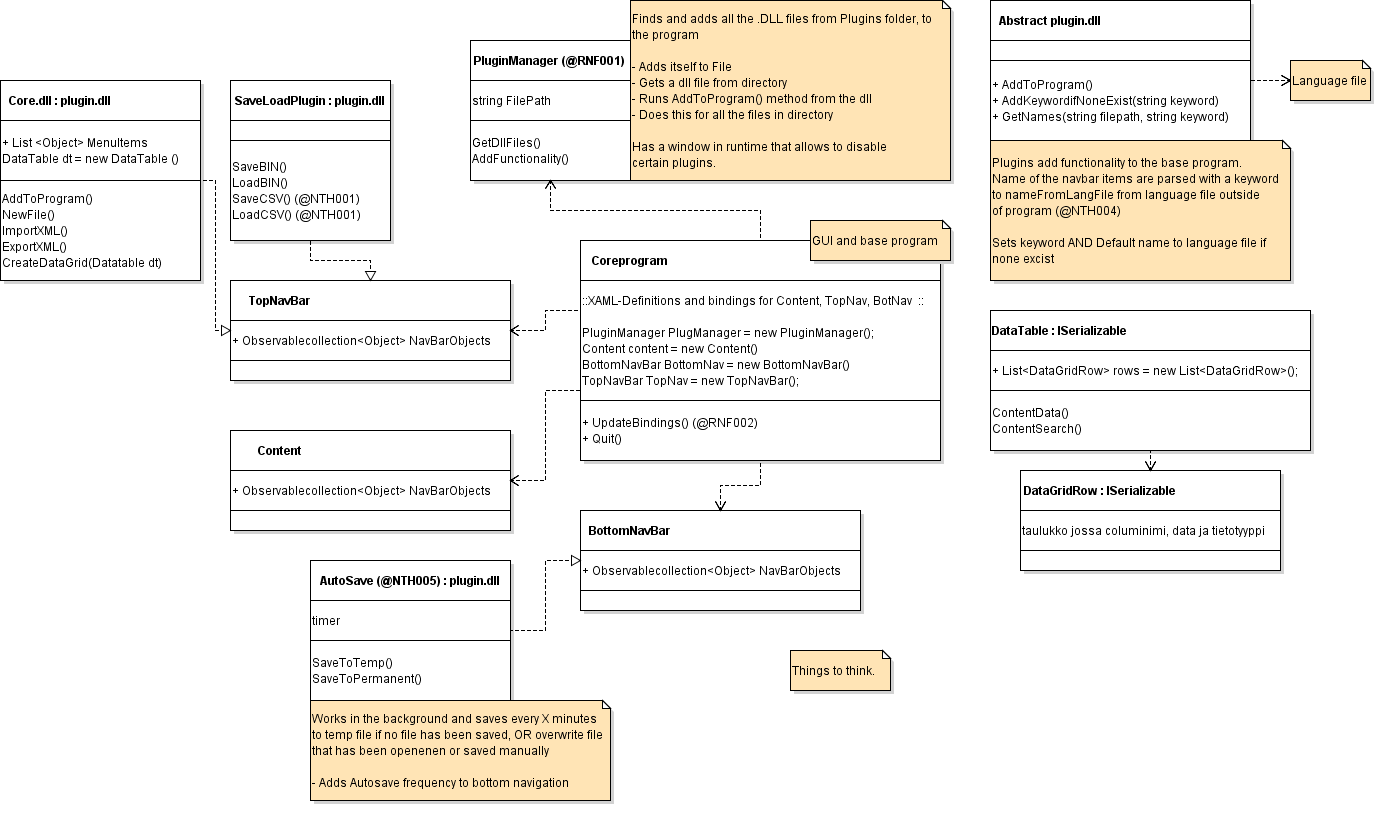


Kuvio 3 Pääohjelma ja plugin suunnitelma

Ohjelman suunnitelma kasvoi hieman isommaksi mitä oli tarkoitus. Opettaja ehdotti CSV tallennus ja lukua sekä Aggregaatti funktioita. Lisäksi itse lisäsin binääri tallennuksen, modulaarisuuden lokalisaation automaattisen tallennuksen ja pari muuta.



Kuvio 4 Jokin satunnainen kohta projekti taulusta.



Kuvio 5 Lopullinen UML Kaavio suunnittelun loppuvaiheilla

# Toteutus vaiheet ja ongelmat

Alku taival oli hyvinkin yksinkertainen, Alustavasti projektin oli tarkoitus olla UWP ohjelma, mutta päätin tehdä WPF sovelluksen, koska kukaan ei käytä enää Windows puhelimia. Tein alustavan WPF layoutin sovellukseen. Toisena tehtävänä oli suunnitella plugin systeemi ohjelmaan, jolla kehittäjät voisivat itse laajentaa ohjelmaa.

## Modulaarisuus

Plugin systeemin kehitykseen meni noin suunnilleen päivä tai kaksi. Löysin todella hyvän resurssin jonka avulla sain tehtyä koodin pätkän joka lataa DLL tiedostot kansiosta. (Creating a simple plugin mechanism.)

Koodi lataa kaikki DLLt kansiosta ja tarkastaa, että ne toteuttavat IPlug Interfacen. Koodi lisää IPlug Interfacen omaavat DLL tiedostot assembly listaan ja ajaa ne ohjelmaan. Yksinkertaistettuna.

## XML Import ja Export

Koulussa koodaus kerroilla tuli puheeksi opettajan kanssa se, miten olin alkujaan ajatellut moni ”käsitteistä” XML tietokantaa ja onko se mahdollista totetuttaa. Opettajan kommenttien pohjalta päätin lähteä yrittämään sitä uudelleen. Viimeksi kun yritin Windows Forms ohjelmassa kesällä 2016 tehdä monikäsitteistä xml tietokanta ohjelmaa, se kaatui tiedoston sisäänlukemiseen ja kirjoittamiseen.

Taistelin päivän kaksi samojen kirjoitus ja luku ongelmien kanssa, kuin edellisessäkin projektissa. Päätin muuttaa ohjelmaa niin, etten käytä valmiita xml kirjoitus kirjastoja, koska monikäsitteisen XML tietokannan kirjoitus siihen muotoon kun luen sen sisään, olisi vaatinut ennalta tietoa XMLn arvoista mitä luet ohjelmaan sisälle. Koska en voi tietää millaisia XML tiedostoja ihmiset luovat ja lukevat sillä sisään. XML scheman luonti oli mahdoton tehtävä, eikä käsitteenä muutenkaan sellainen minkä haluaisin tehdä.

### Oma koodi

Kirjoitin oman koodin XML tiedoston luontiin, tämä oli ainoa vaihtoehto että sain kirjoitettua Datagridin sisällön siihen muotoon, että saan luettua sen sisälle. Tämä XML muoto myös sattuu olemaan muoto, jonka suunnittelin alustavasti oman tietokannan XML pohjaksi.

XML luvussa ja kirjoituksessa tuli todella paljon, painotan TODELLA paljon ongelmia vastaan. Korjasin yhden vian ja toinen ilmestyi. Korjasin toisen vian ja ensimmäinen tuli takaisin. Kyseessä oleva viassa Datagridiin ei pystynyt jostain syystä lisäämään tietoa, kun alkuperäinen dataset oli ladattuna sisään tämä johtui datasetin automaattisesti generoituvasta primääri avaimesta, joka esti soluun kirjoittamisen. Tämän korjaamiseen, käytin Custom datasettiä jonka loin dataset designerissa. Tämä aiheutti ongelmia datasetin käytössä DLL:n lävitse ja oli pakko hylätä.

Metodi kirjoittaa nyt ns. rivikerrallaan datagridin xml tiedostoon. Ongelmaksi muodostui automaattisesti generoidut primääri avaimet, jotka yhdistävät XML relaatiot vanhempiin, jotta kirjoitus saadaan oikeaan muotoon.



Kuvio 6 Malli XML tietokanta

Automaattisesti sisään luetun XML tiedoston relaatiot muodostuvat edelliseen kuvaan siten, että alin pöytä luo avaimen ylempään pöytään. Tämä automaattisesti generoitu primääri avain, luo uuden kolumnin kumpaankin pöytään. Avain jäi kummittelemaan ensin datagridiin ja sen jälkeen XML exporttiin. XML tiedostoa luettaessa primaari avain monistui joka kerta.

XML kirjoituksesta muodostui aikamoinen hirviö, monta sisäkkäistä for silmukkaa sisäkkäin. Jos sisemmän for silmukan sisälle kirjottaisi vielä yhden for silmukan lisää, niin se ei ajaisi sitä. Tämä aiheutui ongelmaksi, koska jotenkin piti pystyä vertaamaan kirjoitettavaa kolumnia, primääri avaimeen, jotta sitä ei kirjoiteta. Lopulta päätin verrata kirjoitettavan kolumnin nimeä primääri avaimen nimeen, että se toimisi. Tämän toteutin ottamalla relaation vanhemman nimen stringiin ja lisäämällä siihen ”\_Id”. Vertasin tätä stringiä sitten kirjoitettavaan kolumnin nimeen.

## Binääri tiedostot

Binaari tallennus ja lataaminen, tapahtui itseasiassa ennemmin, kuin XML ominaisuudet. Projektin hyvin aikaisessa vaiheessa, päätin testata binääri tiedostoon tallennusta ja lukua. Edellisestä projektista muistelin, että isojen XML tiedostojen lukeminen saattaa kestää kauankin aikaa (300Mt = n. 15min ensimmäisessä projektissa kesällä 2016) binaarien tallennus ja luku vastaan xml luku ja tallennus, osoittautuikin todelliseksi ajan säästäjäksi. Testeistä ei jäänyt kuva materiaalia liitettäväksi raporttiin, mutta keskimääräinen ajansäästö binaaria käsiteltäessä oli noin ¾ osa. noin 100Mt:n xml tiedoston avaus kesti koulun koneella noin 45 sekuntia ja sen avaus binaari muodossa, kesti vain noin 10 sekuntia.

Päätin että binääri tiedostojen käyttö ns. projektin hallinnassa olisi todella merkittävä ajansäästö. XML tiedostojen kasvaessa ei tarvitsisi lukea ja kirjoittaa massiivista XML tiedostoa joka kerta. XML kirjoitus tarvii tehdä vain silloin kun tietokanta otetaan käyttöön.

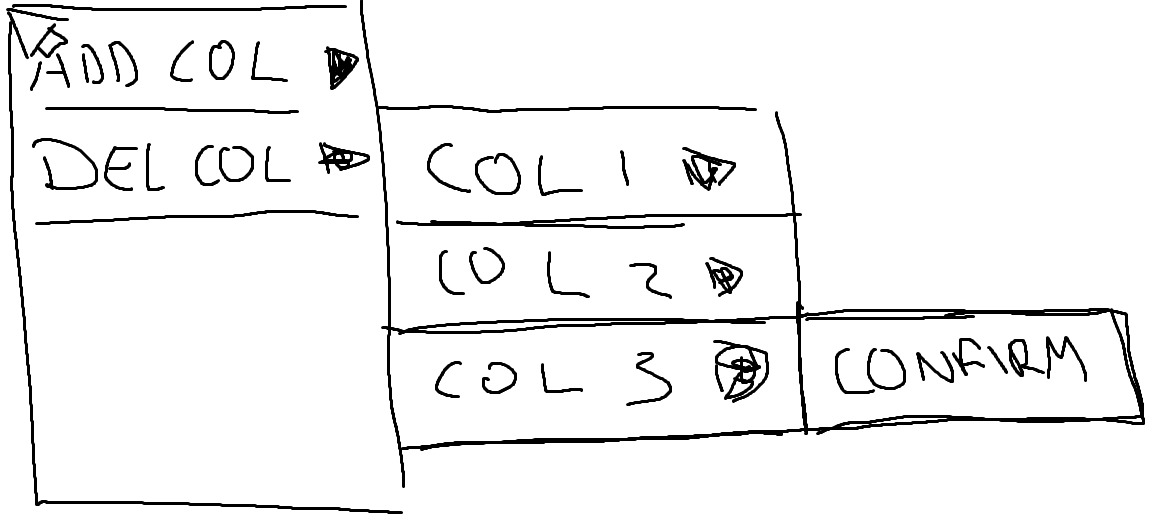
## Perus toiminnot

”New…” toimii periaatteessa niin, että se käynnistää uuden instanssin ohjelmasta. Tämä on suunniteltu eikä pugi, koska pidän sitä tärkeänä, että kahta tietokantaa voi pitää auki samanaikaisesti ja verrata toisiinsa.

”Import from URL” toimii aika yksinkertaisesti. Käytin valmiita kirjastoja lataamaan annetusta nettiosoitteesta XML tiedoston muuttujaan. Tämä muuttuja luetaan XMLreaderilla datasettiin, joka liitetään datagridin datacontextiin.

## Datagrid contextmenu

Lisäsin datagridille context menun jonka avulla käyttäjä voi lisätä, poistaa ja uudelleenimetä kolumneja. Context menun tyyli tuli mieleen, kun hetken tappelin ikkunan lisäyksen kanssa. Ajattelin ensin, että jokainen komento lisää uuden ikkunan joka palauttaa arvon. Tämän toteuttaminen meni yhtäkkiä kovin monimutkaiseksi, joten päätin testata textboxia menuitemin lapseja ja yllätys yllätys se toimi. Päätin tämän pohjalta context menu mallin, jota aion noudattaa IndieXML toteutuksessa nyt ja myöhemmin.



Kuvio 7 Contextmenu wireframe

# Tilanne

Ohjelmasta jäi pari avain asiaa tekemättä huonon ajanhallinnan takia (aikaa mennyt paljon SQL-tietokanta harjoitustyön tekemisessä), jotka olisin halunnut saada valmiiksi jakson aikana. Esimerkiksi realiaikainen haku datasetistä, jota kerkesin vasta tutkimaan. Treeviewiin pitäisi voida lisätä uusia käsitteitä joka luo niille uuden pöydän.

Ohjelmasta jäi puuttumaan myös pari asiaa jotka olisi ollut ihan mukava lisätä, kuten: Automaattinen tallennus ja opettajan mainitsemat CSV tallennus ja luku sekä Aggregaattifunktiot.

Ohjelmaan jäi vielä ongelmia joista osan voisi määritellä pugeiksi ja osan toiminnallisiksi vajauksiksi. Esim. joillain tietokoneilla menut aukeavat vasemmalla

eivätkä oikealla. Try catch event callerissa ei heitä erroria eteenpäin. Väärinmuotoiltu XML tiedosto aiheuttaa ohjelman kaatumisen.

Sain kuitenkin myös paljon aikaan. Opin hirmuisesti ohjelman modularisoinnista, joka oli ennalta täysin vieras asia. Interfacen käytöstä modulaarisuuden yhteydessä.

Sain myös tehtyä usea käsitteisen XML tietokanta ohjelman jota voin laajentaa myöhemmin. Sain binääri tallennukset toimimaan ja XML integraation. Kaikista suurin onnistuminen täytyy olla joko plugin systeemi tai oma XML kirjoitus. Olen aika tyytyväinen molemmista.

# **Pohdinta**

1.0 versioon jää tehtävää vielä CSV toiminnot, Datagrid search ja käsitteiden lisäys treeviewiin. Kriittiset ongelmat jotka kaatavat ohjelman pitää myös korjata. Ulkoasu pitää hioa ja esim. valikoista pitää poistaa kuvake paikat ja varmistaa että valikot aukeavat oikealta puolelta. Haluaisin julkaista ohjelman jossain platformissa, mutta versio 1.0 on saatava ensin. Katsotaan nyt.

# Itsearvio

Mielestäni projekti meni suhteellisen hyvin verraten siihen, että suurin työrupeama tapahtui vasta viimeisellä viikolla. Yksi asia jäi tekemättä mitä pidän kyllä 1.0 versioon välttämättömänä ja se on Search toiminnot. Olisin myös halunnut julkaista ohjelman windows storeen kurssin aikana. 4/5.

Lähteet

Creating a simple plugin mechanism. 21.1.2015. MSDN sivusto. Viitattu 25.4.2017. https://code.msdn.microsoft.com/windowsdesktop/Creating-a-simple-plugin-b6174b62