



TAMPEREEN  
AMMATTIKORKEAKOULU

# ARCHICAD-KIRJASTO DEBEL-KERROSLAT- TIAJÄRJESTELMÄSTÄ

Juho Hakala

Opinnäytetyö  
Huhtikuu 2017  
Rakennustekniikan koulutusohjelma  
Talonrakennustekniikka



## TIIVISTELMÄ

Tampereen ammattikorkeakoulu  
Rakennustekniikan koulutusohjelma  
Talonrakennustekniikka

HAKALA, JUHO:

ArchiCAD-kirjasto Debel-kerroslattiajärjestelmästä

Opinnäytetyö 46 sivua, joista liitteitä 8 sivua  
Huhtikuu 2017

---

Rakennushankkeen suunnittelu toteutetaan useimmiten mallinnusta hyväksikäyttäen. Rakennusosia ja kalusteita sijoittamalla malliin saadaan havainnollistettua lopullista rakennusta jokaiselle ymmärrettävässä muodossa. Visualisoinnilla saadaan aikaan todennukaisia kuvia kohteesta. Tämä edistää ja helpottaa myyntiä ja markkinointia. Tuotevalmistajan kannalta rakennusosat on hyvä olla valmiina, jotta suunnittelijan tehtäväksi jää vain osien sijoittaminen malliin kohteen vaatimusten mukaisesti.

Tässä opinnäytetyössä tehtiin Saint-Gobain Rakennustuotteet oy:n ja Aulis Lundell oy:n valmistamasta Debel-kerroslattiajärjestelmästä objektikirjasto, joka kattaa kyseiseen järjestelmään kuuluvat komponentit. Työssä keskityttiin erityisesti objektien mallintamiseen ja kehittämiseen helppokäyttöisiksi. Lisäksi työssä tehtiin kerroslattian profiileista poikkileikkausprofiilit sekä määrälaskentapohjat edellä mainittujen tuotteiden laskentaa varten.

Lopputuotteena tilaaja sai käyttövalmiin kirjastopakettin, jonka avulla suunnittelijan on mahdollista mallintaa ja määrälaskea Debel-kerroslattiajärjestelmästä koostuva kohde. Työn perusteella tilaajan on mahdollista kehittää vastaavia kirjastopaketteja myös muista järjestelmistään.

## **ABSTRACT**

Tampereen ammattikorkeakoulu  
Tampere University of Applied Sciences  
Degree programme in Construction Engineering  
Option of Building Construction

**HAKALA, JUHO:**  
ArchiCAD-library of a Debel storey floor

Bachelor's thesis 46 pages, appendices 8 pages  
April 2017

---

Design of a building is often done by using modeling. By inserting building elements and furnitures into the model it is possible to visualize the outcome of the building project in a way that is understandable for everyone. Using visualization and rendering results in a realistic model of a project. This advances and simplifies both sales and marketing. In a product manufacturers perspective, it is most suitable if the building elements are already created. Designer is then only required to place the elements in the model according to the project.

In this thesis element parts of the storey floor manufactured by Saint-Gobain Construction products Ltd and Aulis Lundell Ltd were made into an object library. The object library contains the parts that consist of the storey floor. Thesis focused on modeling the objects and developing them as easy to use as possible. Thesis also consists making the section profiles of the storey floor and the templates for quantity survey.

As a result the client received a ready-to-use library package. Library package allows the designer to model and calculate the quantities of a Debel storey floor project. Thesis also allows the client to develop similar library packages from other building systems.

---

Key words: ArchiCAD, GDL, Debel storey floor

## SISÄLLYS

1	JOHDANTO.....	6
2	DEBEL-KERROSLATTIAJÄRJESTELMÄ .....	7
3	3D-MALLINTAMINEN .....	8
	3.1 Yleistä .....	8
	3.2 Graphisoft ArchiCAD.....	8
	3.3 GDL .....	9
	3.4 Trimble Sketchup.....	11
	3.5 Autodesk Inventor.....	12
4	GDL-OBJEKTI .....	14
	4.1 GDL-objektin rakenne .....	14
	4.2 Objektin ohjelmointi .....	16
	4.3 Parametrit.....	17
	4.4 Esiohjelma .....	18
	4.5 2D- ja 3D-ohjelma .....	19
	4.6 Arvolista- ja käyttöliittymäohjelmat.....	21
5	ESIMERKKIOBJEKTI: POLYURETAANIALUSTA .....	23
	5.1 Parametrit.....	23
	5.2 Esiohjelma .....	24
	5.3 3D-ohjelma .....	25
	5.4 Arvolistaohjelma.....	25
	5.5 Käyttöliittymäohjelma .....	26
6	ATTRIBUUTTITIEDOSTOT JA MÄÄRÄLASKENTAPOHJAT .....	29
	6.1 Poikkileikkaukset.....	29
	6.2 Rakennusaineet .....	30
	6.3 Rakennetyypit .....	31
	6.4 Määrälaskentapohjat .....	33
	6.5 Esimerkkikohte .....	34
7	POHDINTA.....	37
	LÄHTEET.....	38
	LIITTEET .....	39
	Liite 1. Polyuretaanialustan ohjelmat .....	39
	Liite 2. Esimerkkikohteen määrälaskennat.....	46

**LYHENTEET JA TERMIT**

2D	Kaksiulotteisen tason lyhenne
3D	Kolmiulotteisen tason lyhenne
BIM	Building Information Model, rakennuksen tietomalli
CAD	Computer Aided Design, tietokoneavusteinen suunnittelu
FEM	Finite Element Method, Elementtimenetelmä lujuuslaskentaan
GDL	Geometric Description Language, geometrinen kuvauskieli
IFC	Industry Foundation Classes, rakennusalan standardi oliopohjaisen tiedon siirtämiseen tietokonejärjestelmästä toiseen
Makro	GDL-objektityyppi, joka on kutsuttavissa muihin objekteihin
Parametri	GDL-objektissa yksittäinen muuttuja
.aat	Attribuuttitiedostomuoto
.gsm	GDL-objektien tiedostomuoto
.stl	Yleisesti käytössä oleva 3D-mallinnuksen tiedostomuoto

## 1 JOHDANTO

Rakentamisessa on viimeisen vuosikymmenen aikana tapahtunut tietotekniikan vallankumous. Suunnittelu on siirtynyt lähes kokonaan tietokonepohjaiseksi. 3D-mallintaminen on tullut pysyväksi osaksi rakentamista. Mallintamisen avulla pystytään jo hyvissä ajoin ratkaisemaan päällekkäisyyksiä ja ristiriitoja, sekä muokkaamaan suunnitelmia tehokkaasti ja nopeasti. Suurissa rakennushankkeissa on tuhansia rakennusosia ja valtava määrä niihin liittyvää tietoa. Näiden hallinnoimiseen on kehitetty BIM-ohjelmia, joihin voidaan sisällyttää rakennusosien lisäksi niihin liittyvää tietoa, jota voidaan jakaa ja jallostaa eteenpäin. Älykäs tietomalli elää koko rakennushankkeen elinkaaren ajan aina hankesuunnittelusta rakennuksen käyttövaiheeseen. Yksi rakennuksen tietomallinnusta varten kehitetyistä ohjelmista on ArchiCAD.

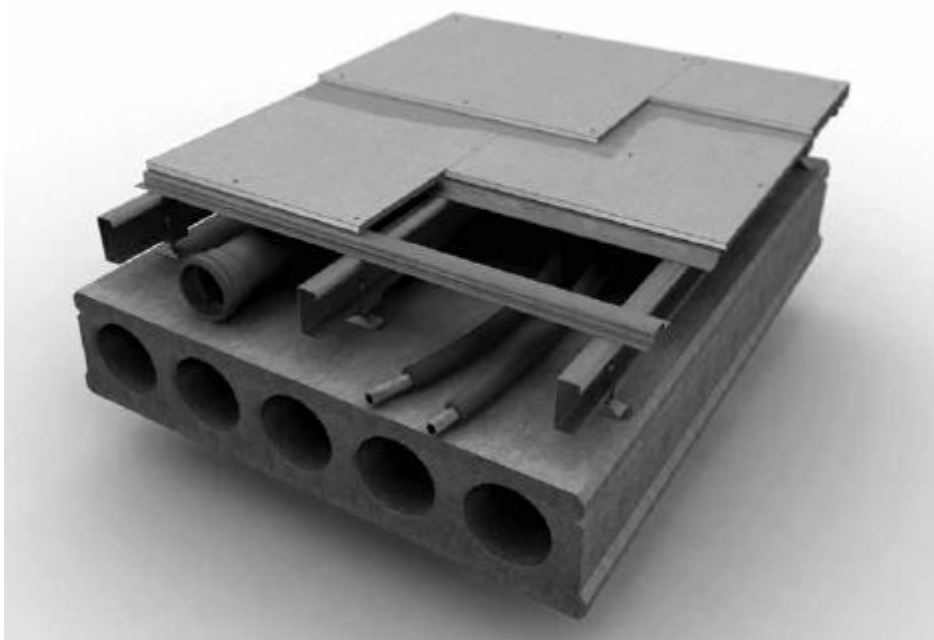
ArchiCAD-ohjelmassa yksittäisiä älykkäitä rakennusosia kutsutaan GDL-objekteiksi. Nämä objektit mahdollistavat omien parametriensa muokkaamisen älykkäällä tavalla rakennusmallin tarvetta vastaavaksi. Objekteihin on mahdollista sitoa tietoa, jota rakennushankkeen osapuolet pystyvät hyödyntämään suunnittelussa ja toteutuksessa. Yleisessä jaossa olevien objektien avulla tuotteiden valmistaja saa näkyvyyttä tuotteelleen.

Opinnäytetyössä tehdään Gyprocin Debel-kerroslattiajärjestelmään tarvittavista komponenteista ArchiCAD-ohjelmaan objektikirjasto ja poikkileikkausprofiilit. Työssä keskitytään älykkäiden GDL-objektien luomiseen, joiden avulla mallintaminen on helppoa ja nopeaa. Työssä tehdään myös valmiit määrälaskentapohjat tiettyyn kohteeseen tarvittaville materiaalimenekeille. Lopputuotteena saadaan suunnittelijalle valmis paketti, jonka avulla on mahdollista mallintaa kerroslattia kohteeseen ja saada määrälaskennan tulokset tulostettua suoraan ohjelmasta.

## 2 DEBEL-KERROSLATTIAJÄRJESTELMÄ

Debel-kerroslattiajärjestelmä on Saint-Gobain Rakennustuotteet oy:n ja Aulis Lundell oy:n kehittämä järjestelmä, joka soveltuu asuin-, majoitus- ja liiketiloihin. Järjestelmä sopii sekä uudis- että korjausrakentamiseen. Kerroslattia koostuu kantavan rakenteen päälle tehtävästä lattiarakenteesta ja erillisestä talotekniikkatilasta. Debel-lattiarakenteen alapuolella on tila vaakasuuntaiselle tekniikalle, kuten sähköjohdoille tai kanaville. Talotekniikkatila on porrashuoneen puolella oleva tila, johon keskitetään huoneistoon tarvittava tekniikka. Siellä sijaitsevat myös kaikki huoneistokohtaiset laitteet. Tilasta lähtevät tarvittavat syötteet kerroslattian sisässä käyttöpisteille. (Saint-Gobain Rakennustuotteet Oy 2015).

Kerroslattiajärjestelmä mahdollistaa tilojen tapauskohtaisen muunneltavuuden eivätkä eri tilojen ratkaisut ole sidoksissa toisiinsa. Huollettavuus paranee, koska kaikki tekniikka sijaitsee keskitetysti yhdessä tilassa. Huoltaminen ei myöskään häiritse huoneistojen käyttöä. Putkivedot voidaan toteuttaa järjestelmän avulla huoneistokohtaisesti käyttäjien toiveiden mukaisesti. Järjestelmä kokonaisuudessaan on ns. kuivarakentamista eli rakenteiden päälle ei tule valu- tai tasoitekerrosta. Tämä nopeuttaa rakentamista, koska kuivumisaikaa ei ole ja kosteusriskit vähenevät. (Aulis Lundell Oy, 3-4).



KUVA 1. Debel-kerroslattia (Gyproc käsikirja, 371)

## 3 3D-MALLINTAMINEN

### 3.1 Yleistä

3D-mallinnuksella tarkoitetaan yksinkertaisuudessaan sitä, että luodaan kaksiulotteisesta kuvasta kolmiulotteinen malli. Kolmiulotteisella mallilla on korkeutta, leveyttä ja syvyyttä. Kaksiulotteisessa kuvassa syvyys puuttuu. Vielä 1970-luvulla koneiden ja laitteiden kokoonpanopiirustukset piirrettiin paperille pöydällä ja osakokoonpanot tehtiin näiden piirustusten pohjalta. Muutoksia tehdessä jokaisen osan sopivuus pitää tarkistaa erikseen. Tämä on kankeaa ja aikaa vievää. Tietokoneiden yleistyminen 1980-luvulla johti suunnittelun siirtymiseen paperilta näytölle. Mallinnuksen avulla osien törmäyskohdat saadaan helposti esille. Lisäksi älykkäissä ohjelmistoissa riittää, että muuttaa muuttuneen osan mittoja ja tähän kytketty geometria muuttuu lisäksi muissa tähän liittyneissä osissa ja päivittyy piirustuksiin.

Rakentamisessa suunnittelu on kehittynyt kaksiulotteisesta piirtämisestä 3D-mallinnuksen kautta nykyisiin älykkäisiin tietomallinnusohjelmiin. Rakennuksen tietomalli eli lyhennettynä BIM tarkoittaa rakennuksen kolmiulotteista mallia, joka sisältää lisäksi tarpeellista lisäinformaatiota rakennuksesta ja sen osista. Näiden tietojen pohjalta voidaan muodostaa esimerkiksi kustannusarvioita ja erilaisia simulaatioita. Rakennuksen tietomallin tavoitteena on, että se on käytössä koko rakennuksen eliniän ajan ja sitä voivat hyödyntää niin suunnittelijat kuin rakennuksen lopulliset käyttäjät.

### 3.2 Graphisoft ArchiCAD

ArchiCAD on unkarilaisen Graphisoft-yrityksen kehittämä ja ylläpitämä erityisesti rakennussuunnitteluun suunnattu rakennuksen tietomallinnusohjelma. Graphisoft perustettiin yrityksenä vuonna 1992. Graphisoft esitteli ensimmäisen ArchiCAD ohjelman version vuonna 1984 ollen arkkitehtisuunnittelun edelläkävijä. Heti ensimmäisestä versiosta alkaen ohjelmassa pystyi luomaan objekteja GDL-ohjelmointikielen avulla. Ensimmäiset kuusi versiota ohjelmasta julkaistiin vain Applen Macintosh käyttöjärjestelmälle. Ensimmäinen Windows-versio, ArchiCAD 4.16, julkaistiin vuonna 1993. (MacKenzie & Rendek, 11). Uusin julkaistu versio, ArchiCAD 20, julkaistiin vuonna 2016. Järjestyksessään

se on 28. versio ohjelmasta. Nykyään uusi versio tulee vuosittain. Suomessa ArchiCADin jakelusta ja tuesta vastaa M.A.D. Oy.

ArchiCAD soveltuu rakennushankkeen joka vaiheeseen aina hankesuunnittelusta kiinteistönhallintaan. ArchiCADissa luodaan yksi tiedosto, joka sisältää itse 3D-mallin sekä kaikki piirustukset. Malliin tehdyt muutokset päivittyvät automaattisesti myös piirustuksiin. Tämä vähentää selvästi virheitä verrattuna muihin työmenetelmiin. Visualisointi on myös viety pitkälle. ArchiCAD-mallista pystyy tuottamaan virtuaalimaailman, jossa pystyy liikkumaan ja havainnollistamaan suunnittelun kohdetta helposti.

### 3.3 GDL

ArchiCAD-objektit luodaan GDL-ohjelmointikielillä. GDL, Geometric Description Language, on oma ohjelmointikielensä, joka on kehitetty BASIC-ohjelmointikielen pohjalta. GDL on Graphisoftin kehittämä ArchiCADia varten, mutta se on yhteensopiva muiden tietomallinnusohjelmien kanssa. GDL-objektien rakenne koostuu eri komentolistauksista, joiden perusteella ArchiCAD muodostaa objektin kaksi- ja kolmiulotteisen mallin. GDL-objekti voi sisältää myös tekstitietoa, kuten tietoa valmistajasta tai tarvittavaa materiaalitietoa määrälaskentaa varten. GDL on ohjelmointikielenä avoin, eli kuka tahansa pystyy luomaan, kehittämään ja jakamaan ArchiCADin eri rakenneosia, kuten objekteja ja rakennetyyppejä, käytännössä rajattomasti. Tämä edistää suunnittelua. Pelkästään Graphisoftin sivuilla onkin jaossa tuhansia valmiita objekteja.

GDL-objektien luonti tapahtuu syöttämällä komentosarjoja eri ohjelmien sisälle. Objektin sijoitusvaiheessa ArchiCAD lukee komentosarjat rivi kerrallaan ja toteuttaa komennot. Eri komentojen avulla saadaan aikaan riippuvuuksia ominaisuuksien välille tai esimerkiksi tietty asia toistumaan. Pääasiallisesti objektin luomiseen on kolme tapaa; ohjelmointi, ArchiCAD-työkalujen käyttö ja mallintaminen erillisellä ohjelmalla.

GDL-ohjelmoinnilla luotujen objektien hyöty tulee siinä, että objekteista voidaan tehdä älykkäitä, parametrisoituja ja helppokäyttöisiä. Ohjelmoija vaikuttaa koko objektin luomiseen alusta loppuun asti, jonka takia objekti sisältää vain sille tarpeellista ohjelmointitekstiä. Samasta syystä objektit ovat myös kooltaan pienempiä kuin mallintamalla tehdyt objektit. Tämän merkitys korostuu suuren kokoluokan projekteissa, jotka sisältävät ison

määrän objekteja. Pidemmän koodin lukeminen vie ohjelmalta enemmän aikaa, ja kulunut aika luonnollisesti moninkertaistuu aina kun objektien määrä kasvaa. Vastaavalla tavalla myös renderointi ja visualisointi hidastuu. Nykypäivänä tietokoneiden laskentateho on kasvanut merkittävästi, joten objektien koolla ei ole niin suurta merkitystä kuin muutamia vuosia sitten. Objektien koko on kuitenkin hyvä pitää mielessä, loi objektit millä menetelmällä tahansa. Ylisuuria objekteja ei kannata tarkoituksellisesti tehdä.

GDL on ohjelmointikielenä käytössä vain ArchiCAD-ohjelmassa, joten sen levinneisyys on rajallinen. Tämä aiheuttaa suurimmat ongelmat objektien luomisessa ohjelmoimalla. Ohjeita on vain muutama, lähes poikkeuksetta englanniksi. Objektien koodaaminen tekstimuodossa altistaa helposti virheille. Käytännössä yksikin pilkku väärässä kohdassa tarkoittaa sitä, että koko objektin ohjelma lopettaa toiminnan. Kommentosarjalle on olemassa koodin tarkistamiseen työkalu, mutta sekin antaa hyvin vähän tietoa virheen korjaamiseksi. Työkalu ilmoittaa rivin, jossa virhe sijaitsee sekä tietoa virheestä, kuten mitä käskyä on käytetty. ArchiCAD ei kuitenkaan ilmoita rivinnumeroita ohjelmissa, joten virheen paikantaminen vie aina jonkin verran aikaa. Lisäksi hyvin usein jää käyttäjän tehtäväksi selvittää tarkemmin mitä kyseenomaisessa komentosarjassa on pielessä.

GDL-objekteja pystyy luomaan myös ArchiCAD-ohjelman valmiita työkaluja käyttäen. Tässä tapauksessa haluttu objekti piirretään valmiiksi mallissa, jonka jälkeen se tallennetaan haluttuun kirjastoon objektina. Tämän jälkeen tehtyä objektia voidaan käyttää kuten mitä tahansa valmista objektia projektissa. Esimerkkinä voidaan ottaa kattoristikko. Tämä voidaan piirtää palkki-työkalulla oikeaan kokoisena ja muotoisena. Valmis kattoristikko tallennetaan objektina, jonka jälkeen sitä on helppo monistaa ja tarvittaessa käyttää tulevaisissa projekteissa. Objektien luonti työkalujen avulla kannattaa tilanteessa, jos geometrialtaan yksinkertainen objekti tarvitaan nopeasti malliin. Mikäli haluttu objekti on monimuotoinen, kuluu sen piirtämiseen runsaasti aikaa eikä objektin piirtäminen välttämättä edes onnistu.

Kolmas vaihtoehto on mallintaa objekti erikseen siihen tarkoitettulla ohjelmalla, jonka jälkeen valmis rakenne tuodaan stl-muotoisena ArchiCAD-ohjelmaan GDL-objektiksi. Markkinoilla on nykyään useita 3D-mallinnukseen tarkoitettuja ohjelmia. Näiden avulla monimuotoisten objektien luominen on vaivatonta. Tämä säästää aikaa työläältä ohjelmoinnilta ja ehkäisee virheiden syntymistä. Valmiiksi tehtyyn objektiin voidaan muokata

ArchiCADin puolella tarvittavat parametrit sekä valikot ja mahdolliset määrälaskentatiedot.

Muutamia yleisesti käytettyjä 3D-mallinnusohjelmia ovat:

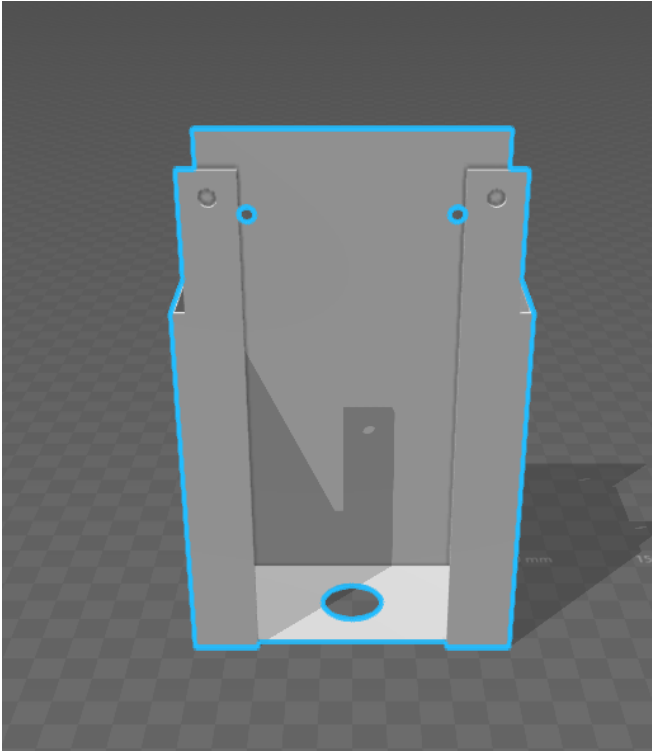
- Trimble Sketchup
- Autodesk Inventor
- Autodesk Fusion
- Solidworks

Tässä opinnäytetyössä esitellään edellä olevasta listasta kaksi ensimmäistä.

### **3.4 Trimble Sketchup**

Sketchup on 3D-mallinnusohjelma, jonka kehitti alun perin @Last Software. Nykyään ohjelman omistaa Trimble Inc. Ohjelman kehitti muutama 3D-mallinnuksen veteraani, jotka halusivat ohjelman, jolla pystyisi helposti esittämään suunnitelmat kolmiulotteisena. Nykyään ohjelmaa käytetään paitsi yksinkertaiseen mallinnukseen myös arkkitehti- ja insinöörisuunnittelussa. (Chopra 2011, 1). Sketchup-ohjelmasta on tarjolla sekä ominaisuuksiltaan rajatumpi ilmaisversio että maksullinen täysiversio.

Ohjelman käyttö on aloittelijallekin helppoa. Yksinkertaisuudessaan ohjelmassa piirretään tasoon viiva-työkalulla haluttu kuvio. Kuvioista saadaan kolmiulotteinen vetämällä sitä haluttuun suuntaan, jolloin kuvio nousee tasosta. Aukkojen leikkaaminen tapahtuu käänteisesti. Haluttu leikkauskuvio piirretään jo olemassa olevaan malliin, jonka jälkeen aukko tehdään vetämällä halutun syvyinen aukko kohteeseen. Kuvassa 2 esitetään Debelkerroslattiajärjestelmän korotusjalka, joka on ensin mallinnettu Sketchup-ohjelmalla ja tulostettu stl-muotoisena.



KUVA 2. Korotusjalka mallinnettuna Sketchup-ohjelmalla

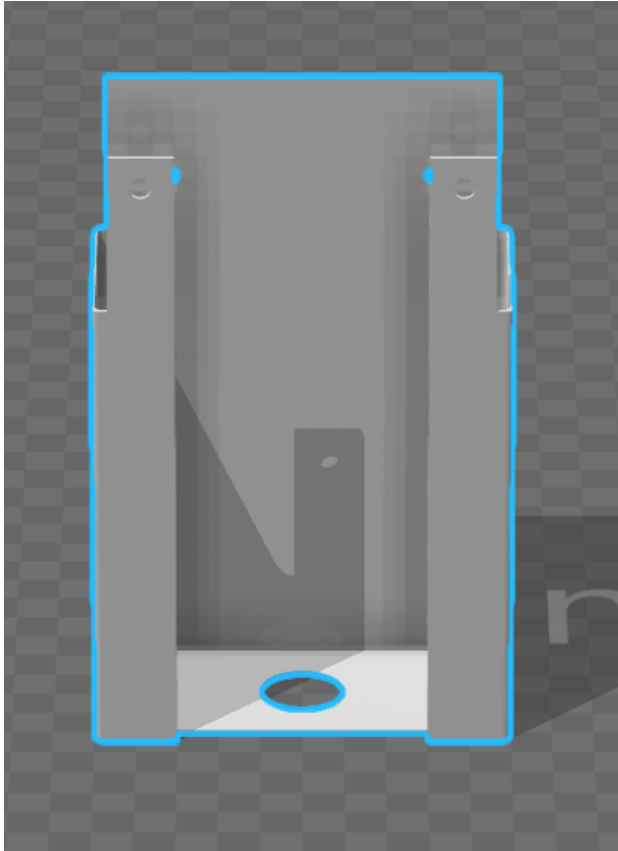
Sketchup soveltuu helpon käytettävyyden ansiosta erinomaisesti ensimmäiseksi ohjelmaksi 3D-mallinnuksen harjoitteluun sekä yksinkertaisten mallien ja objektien tekemiseen. Esimerkiksi rakennusten karkea mallinnus onnistuu nopeasti ja helposti Sketchupin avulla.

### 3.5 Autodesk Inventor

Autodesk Inventor on 3D CAD ohjelma, jota käytetään renderointiin, tuotesuunnitteluun ja simulointiin. Sen ensimmäinen versio julkaistiin vuonna 1999 ja uusin, järjestyksessään 22 versio julkaistiin vuonna 2016. Ohjelma on tarkoitettu pääasiallisesti mekaniikkasuunnitteluun ja sillä onkin mahdollista simuloida esimerkiksi tehtaiden liukuhihnojen toimintaa. Ohjelman vahvuutena on myös valmistajan laaja tuoteperhe, joiden kanssa tuote on yhteensopiva.

Inventor hyödyntää parametristä piirremallinnusta. Parametrisuus tarkoittaa sitä, että ohjelmassa mallinnettujen osien mittoja voidaan muuttaa missä vaiheessa tahansa ja koko kohteen geometria muuttuu vastaavasti. Piirremallinnus tarkoittaa vuorostaan sitä, että malli muodostuu eri piirteistä. Aluksi tehdään peruspiirre, johon lisäämällä uusia piirteitä

saadaan aikaiseksi lopullinen malli. (Hietikko 2007, 23). Kuvassa 3 esitetään korotusjalka mallinnettuna Inventor-ohjelmalla ja tulostettuna stl-muotoisena.



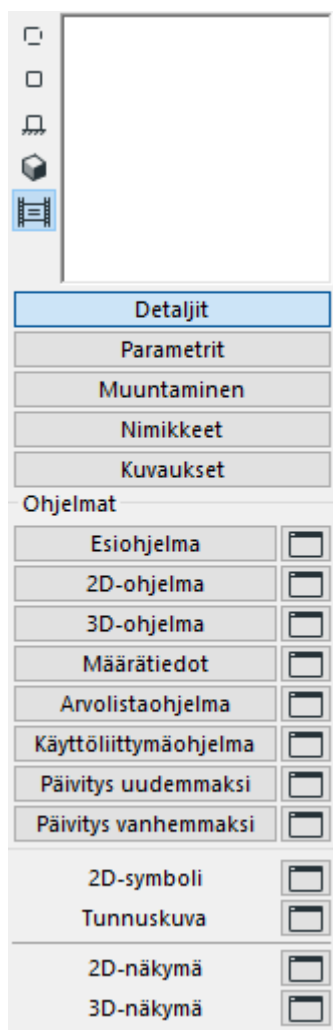
KUVA 3. Korotusjalka mallinnettuna Inventor-ohjelmalla

Inventor on huomattavasti monikäyttöisempi mallinnusohjelma kuin Sketchup. Ohjelman avulla onnistuu monimuotoisten kokoonpanojen suunnittelu ja mallintaminen. Ohjelmalla on mahdollista tehdä simulaatioita koneiden toiminnasta ja sillä onnistuvat myös erilaiset analyysit, kuten FEM-laskenta.

## 4 GDL-OBJEKTI

### 4.1 GDL-objektin rakenne

GDL-objekti muodostuu eri ohjelmista ja tietoikkunoista, jotka yhdessä kokonaisuutena muodostavat valmiin objektin. GDL-objektin tunnistaa .gsm-tiedostopäätteestä. ArchiCAD lukee GDL-objektin ohjelmat aina tietyssä järjestyksessä, joten ennen objektin luomista on tärkeää tietää ohjelmien ja tietoikkunoiden merkitys. Tietyt ohjelmat ja tietoikkunat kommunikoivat keskenään ja onkin tärkeää miettiä mihin päättää komennon kirjoittaa. Tällä tavalla säästää aikaa ja objektista tulee kevyempi luettava ohjelmalle. Kuvassa 4 esitetään GDL-editorin rakenne ja taulukossa 1 esitetään tietoikkunoiden ja ohjelmien selitteet.



KUVA 4. GDL-editorin rakenne

TAULUKKO 1. Komentoikkunat ja niiden selitteet

Komentoikkuna	Selite
<b>Esikatselu-ikkuna</b> 	Sisältää objektin esikatselun 2D-symbolin, 2D- ja 3D -näkyvän, sekä tunnuskuvan.
<b>Detaljit</b>	Sisältää tietoa objektista sekä objektin alatyypin valinnan.
<b>Parametrit</b>	Objektin muokkaamiseen tarvittavat parametrit
<b>Muuntaminen</b>	Sisältää objektin versiohistorian
<b>Nimikkeet</b>	Sisältää objektin luetteloinnin ja määrälaskennan asetuksia
<b>Kuvaukset</b>	Luetteloinnin ja määrälaskennan kuvaukset
<b>Esiohjelma</b>	Pääkomentoikkuna, joka luetaan ennen jokaista komentoikkunaa.
<b>2D-ohjelma</b>	Objektin 2D geometria ja tartuntapisteet
<b>3D-ohjelma</b>	Objektin 3D-geometria ja tartuntapisteet
<b>Määrätiedot</b>	Määrälaskennan toteutus ja kuvaus
<b>Arvolistaohjelma</b>	Komentoikkuna parametrien määrittämiseen ja muokkaamiseen. Kommunikoii parametrit-komentoikkunan kanssa
<b>Käyttöliittymäohjelma</b>	Komentoikkuna tarkemman dialogisivun luomista varten. Valikko voi sisältää kuvia, painikkeita ja selitteitä objektista
<b>Päivitys uudemmaksi</b>	Määrittely objektin päivittämiseksi uudempaan ohjelmaversioon
<b>Päivitys vanhemmaksi</b>	Määrittely objektin päivittämiseksi vanhempaan ohjelmaversioon
<b>2D symboli</b>	Objektin tasokuva, jonka voi piirtää käsin tai tallentaa suoraan kuvasta. Näkyvissä vain, jos 2D-ohjelmaa ei ole.
<b>Tunnuskuva</b>	Objektin ”mainoskuva”, joka näkyy objektia valittaessa objektikirjastossa

<b>Komentoikkuna</b>	<b>Selite</b>
<b>2D-näkymä</b>	2D-ohjelma perusteella määritelty objektin tasokuva
<b>3D-näkymä</b>	3D-ohjelman perusteella määritelty objektin 3D-näkymä

## 4.2 Objektin ohjelmointi

GDL-ohjelmoinnin historia lähtee liikkeelle ydinvoimaloiden putkisuunnittelusta 1980-luvulla. Tuolloin suunnittelu suoritettiin kokonaisuudessaan kirjoittamalla tekstimuotoisia komentosarjoja ilman hiirtä tai käyttöliittymiä. Tästä kehitettiin lopulta arkkitehtisuunnittelua varten ArchiCAD-ohjelma, mutta GDL säilyi ohjelmointikielenä objekteille. (Nicholson-Cole 2000, 15). Yhä edelleen GDL-objektin näkyvä rakenne muodostuu ohjelmiin kirjoitetuista komennoista. GDL-ohjelmointi perustuu neljään osa-alueeseen, jotka on esitetty taulukossa 2.

TAULUKKO 2. GDL-ohjelmoinnin osa-alueet (Nicholson-Cole 2000, 15)

<b>GDL-komennot</b> 2D- ja 3D-muotojen luominen, esimerkiksi neliö, kolmio, kuutio ja lieriö	<b>3D-rakenne</b> Objektin kokoamiseen tarvittavien muotojen analysointi ja määrittäminen.
<b>Ohjelmointi</b> Komentosarjojen kirjoittaminen järjestelmällisellä ja tehokkaalla tavalla	<b>Käyttöliittymä</b> Objektien parametrien ja niiden valikoiden määrittäminen vastaamaan objektin ja projektin tarvetta

Ennen objektin ohjelmoinnin aloittamista on kannattavaa käyttää hetki aikaa objektin toteutuksen suunnitteluun. Muutamia kysymyksiä joita kannattaa esittää itselleen on listattu seuraavaksi:

- Minkä tyyppinen objekti on?
- Millä tavalla objekti toteutetaan?
- Tarvitaanko käyttäjälle muokattavia parametrejä?
- Lukitaanko objektin koko?
- Tarvitaanko objektiin käyttöliittymää?

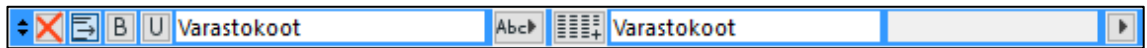
Hyvään ohjelmointitapaan kuuluu olennaisena osana kirjoitetun koodin selittäminen. GDL-kielessä tämä tapahtuu kirjoittamalla komentorivin alkuun huutomerkki. ArchiCAD jättää tällöin kyseisen rivin lukematta. Tällöin seuraava käyttäjä pääsee nopeasti selville siitä, mitä kyseisellä komentosarjalla saadaan aikaiseksi. Kommentointi auttaa myös objektin alkuperäistä tekijää, koska objektin tekijä joutuu jatkuvasti selittämään mitä ollaan tekemässä ja miksi. Tästä seuraa järjestelmälliset ja tehokkaat komentosarjat. Komentosarjojen luettavuuden parantamiseksi on muutama yleisesti käytössä oleva nyrkkisääntö. Varsinaiset komennot tulee kirjoittaa isoilla kirjaimilla ja loput tekstistä kuten parametrit ja kommentit pienillä kirjaimilla. Tämä auttaa myös hahmottamaan komentosarjoja paremmin. Komentosarjoissa kannattaa käyttää sisennyksiä selventämässä eri komentosarjojen eroja.

### 4.3 Parametrit

Parametrit-komentoikkunassa määritellään GDL-objektin muuttujat ja ominaisuudet. Jokaisella kolmiulotteisella kappaleella on aina X-, Y- ja Z-mitta. GDL-objektissa näille vakioparametreille on annettu nimitykset A, B ja ZZYZX. Jälkimmäisin saattaa aiheuttaa hämmennystä, mutta sen nimelle on selkeä selitys. ArchiCADin alkuaikoina Z-muuttuja oli jo käytössä muissa ominaisuuksissa ohjelmassa. Objektin korkeudelle oli keksittävä jokin toinen nimitys. Kehitystiimi keksi nimetä parametrin Mojaven autiomaassa Yhdysvalloissa sijaitsevan pikkukaupungin, Zzyzx:n, mukaan. Tämä nimitys on säilynyt tähän päivään asti. (Nicholson-Cole 2000, 60). Neljäs vakioparametri GDL-objekteilla on valinta, joka määrittää, näytetäänkö 2D-tartuntapisteet myös 3D-maailmassa. Detaljit-komentoikkunassa määritetyllä objektin alatyypillä vaikutetaan objektin parametreihin. Esimerkiksi valittaessa alatyypiksi pöytä ArchiCAD tarjoaa automaattisesti objektille runsaasti parametreja, kuten tason ja jalkojen materiaalien valikon. Parametreja voi myös luoda rajattomasti käsin. Parametrit-käyttöliittymä kommunikoi ensisijaisesti arvolista-ohjelman kanssa, jonka avulla voidaan määrittää parametreille arvoja ja ehtoja.

Parametrien avulla objektille voidaan myös määrittää otsikoita ja valikkoja objektin valintaikkunaan. Yleisimpiä valikkoja ovat objektin materiaalit ja objektin kokoon liittyvät parametrit. Uuden parametrin luomisen jälkeen sille määritetään tyyppi. Parametri voidaan piilottaa, jolloin sitä ei näytetä käyttäjälle valintaikkunassa. Sarkain-napin aktivoi-

malla parametri saadaan alistettua, jolloin se näkyy edeltävän päätasoisien parametrin ali-valikossa, joka voidaan avata ja sulkea. Tätä kannattaa hyödyntää, jotta valintaikkuna pysyy siistinä. Parametrin teksti voidaan myös lihavoida. Uniikki-napin aktivointi estää kyseisen parametrin arvojen siirtymisen toiseen objektiin. Kuvassa 5 esitetään esimerkkiparametri ”Varastokoot” ja edellä mainitut näppäimet. Parametrin tyyppi on teksti ja sen arvot määritetään arvolistaohjelmassa.



KUVA 5. Esimerkkiparametri

#### 4.4 Esiohjelma

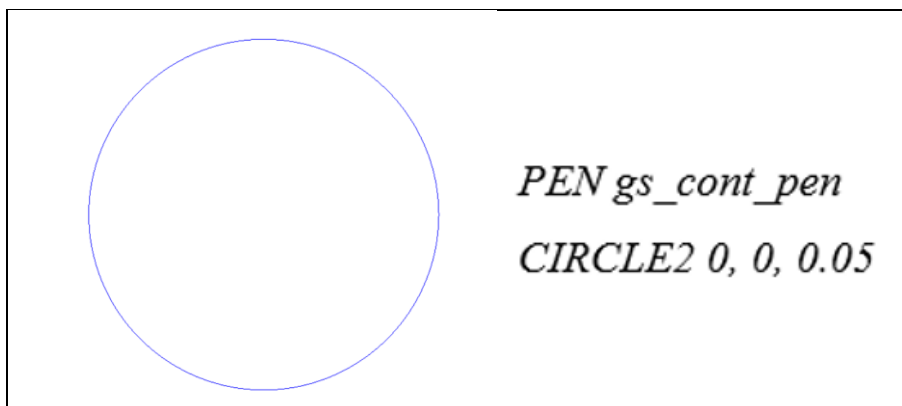
Esiohjelma on GDL-objektin ”pääohjelma”. ArchiCAD lukee esiohjelman aina ennen muiden ohjelmien suorittamista. Esiohjelmaan kannattaa taten sisällyttää sellaiset objektin ominaisuudet, joiden haluaa toistuvan. Muussa tapauksessa haluttavat ominaisuudet tulisi kirjoittaa jokaiseen ohjelmaan, joissa niiden haluaa toimivan. 2D- ja 3D-ohjelma eivät ole suoraan riippuvaisia toisistaan, mutta esiohjelman avulla objektille voidaan määrittää komentoja, jotka ovat käytössä sekä 2D:ssä että 3D:ssä. Taulukossa 3 esitetään muutamia esiohjelmassa yleisesti käytettäviä komentoja.

TAULUKKO 3. Esiohjelman yleisiä komentoja

Komento	Selite
Call	Makron kutsuminen objektiin
Dim	Matriisin luominen
If – then - else	Ehtokomento, jonka ollessa tosi ohjelma tekee tietyn asian ja ollessa epätosi ohjelmaa tekee tietyn asian
Endif	Päättää ehtokomennon
End	Päättää ohjelman lukemisen

## 4.5 2D- ja 3D-ohjelma

2D-ohjelmalla määritetään objektin ulkoasu tasonäkymissä. Mikäli tässä ohjelmassa ei ole komentoja, objektin tasokuvana näkyy 2D-symbolin kuva. Samainen kuva saadaan näkyviin myös Fragment2-komennolla. Project2-komennolla objektin tasokuvaksi saadaan piirrettyä kaikki mitä 3D-näkymän projektiossa näkyy. Varsinkin monimuotoisissa objekteissa tämä saattaa aiheuttaa sekavan tai epäselvän tasokuvan. Suositeltavin tapa on määrittää komentosarjojen avulla objektin tasonäkymä. 2D-ohjelmassa objektille voidaan määrittää tartuntapisteitä, joiden avulla objektia voidaan venyttää ja siirtää. Kuvassa 6 esitetään esimerkkikomentosarja, jonka avulla saadaan piirrettyä ympyrä ja sen tasonäkymä. ”Pen”-komennolla määritetään viivatyyppi, jolla kuvio piirretään. Tässä tapauksessa se on määritetty noudattamaan ”gs\_cont\_pen”-parametriä. Parametrissa on valittuna sininen väri, joten ympyrä piirtyy sinisenä. ”Circle2”-komennolla piirretään itse ympyrä. Siinä määritetään ensin x- ja y-koordinaatit, ja lopuksi ympyrän säde.



KUVA 6. Esimerkkiympyrä ja sen komentosarja

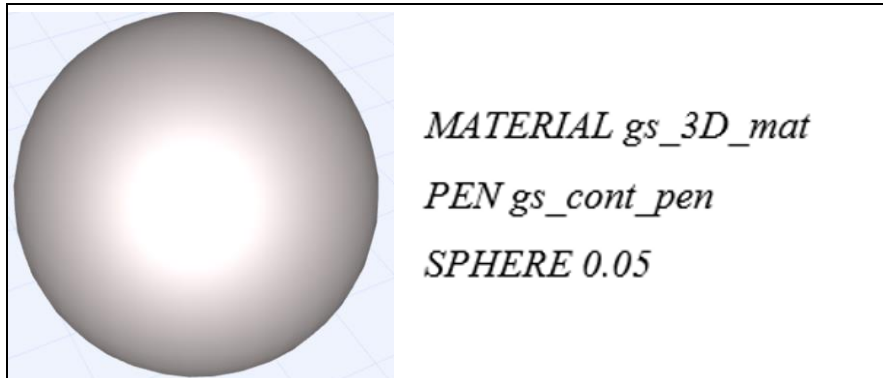
Taulukossa 3 esitetään muutamia yleisesti käytettäviä 2D-komentoja.

TAULUKKO 3. 2D-ohjelman yleisimpiä komentoja

<b>Komento</b>	<b>Selite</b>
Line2	Viivan piirtäminen
Rect2	Suorakulmion piirtäminen
Poly2	Monikulmion piirtäminen
Circle2	Ympyrän piirtäminen

3D-ohjelma on mallinnuksen näkökulmasta ajatellen se tärkein ohjelma. Siinä määritellään objektin geometria, materiaalit ja tartuntapisteet. Yksinkertaisen objektin osalta voidaan lähes kaikki työ tehdä 3D-ohjelman sisällä. Monimuotoisemman objektin osalta on

tehokkaampaa hyödyntää muita ohjelmia. Kuvassa 7 esitetään esimerkkipallon piirtämiseksi tarvittava komentosarja ja sen lopputulos 3D:ssä. ”Material”-komennolla määritetään objektin materiaali, tässä tapauksessa ohjelma hakee sen ”gs\_3D\_mat”-parametristä. ”Pen”-komento toimii kuten 2D:ssä. ”Sphere”-komennolla saadaan piirrettyä pallo ja sille haluttu säde.



KUVA 7. Esimerkkipallo ja sen komentosarja

Taulukossa 4 esitetään muutamia yleisesti käytettäviä 3D-komentoja. Kappaleiden kokoa saadaan muokattua komentojen jälkeen kirjoitetuilla arvoilla.

TAULUKKO 4. 3D-ohjelman yleisimpiä komentoja

<b>Komento</b>	<b>Selite</b>
Block	Kuutio
Cylind	Sylinteri
Sphere	Pallo
Ellips	Puolikas ellipsoidi
Cone	Kartio
Prism	Prisma

#### 4.6 Arvolista- ja käyttöliittymäohjelmat

Arvolistaohjelman avulla voidaan määrittää tarkemmin haluttuja parametreja. Arvolistaohjelma kommunikoi siis parametrit-ohjelman kanssa. Arvolistaohjelman avulla on mahdollista myös piilottaa ja lukita parametreja, joiden ei haluta olevan objektin käyttäjän saatavilla. Taulukossa 5 esitetään muutamia yleisimpiä arvolistaohjelman komentoja.

TAULUKKO 5. Arvolistaohjelman komentoja

Komento	Selite
Values	Parametrien arvojen määrittäminen
Range	Parametrien raja-arvojen määrittäminen
Lock	Lukitsee halutun parametrin
Hideparameter	Piilottaa halutun parametrin

Käyttöliittymäohjelman avulla objektille on mahdollista luoda visuaalisesti näyttävämpi ja käyttäjäystävällisempi käyttöliittymä, jolla voidaan kontrolloida parametreja. Käyttöliittymä ei ole objektin toiminnan kannalta välttämätön, mutta kunnollisen käyttöliittymän tekeminen helpottaa objektien parametrien hallintaa. Käyttöliittymän komentosarjat erottaa muista komennoista UI\_-etuliitteestä. Käyttöliittymässä objektin eri osien määrittämiselle on mahdollista tehdä omia sivuja, jolloin käyttäjän on helppo hahmottaa mitä osaa objektista määritellään. Esimerkiksi pöydän osalta voisi käyttöliittymässä olla omat sivut jalkojen ja pöytälevyjen, sekä määrätietojen ominaisuuksille. Vertauksen vuoksi ilman käyttöliittymää nämä ominaisuudet olisivat allekkain objektin asetuksissa. Käyttöliittymä on objektin osalta ainut paikka, jonne pystyy lisäämään painikkeita esimerkiksi valmistajien internetsivuille. Toimivan ja helppokäyttöisen käyttöliittymän tekeminen tyhjästä vie aikaa, mutta lopputulos on sen arvoinen. Lisäksi kerran kunnolla tehty käyttöliittymä on helppo kopioida pääpiirteittäin seuraaviin objekteihin, poistaen siitä vain tarpeettomat osaset. Taulukossa 6 esitetään käyttöliittymäohjelmassa yleisimmin käytettyjä komentoja.

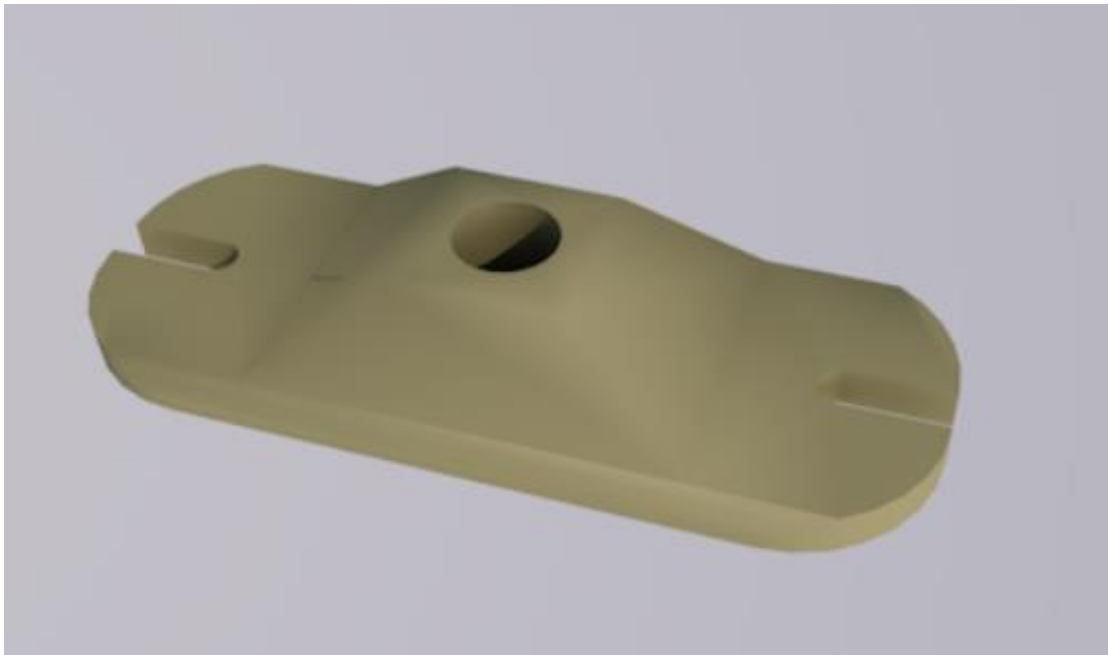
TAULUKKO 6. Käyttöliittymäohjelman komentoja

<b>Komento</b>	<b>Selite</b>
UI_Dialog	Valintaikkunan otsikon määrittäminen
UI_Page	Sivunumeron määrittäminen
UI_Style	Tekstin tyylin määrittäminen
UI_Button	Näppäimen määrittäminen sivulle
UI_Pict	Kuvan määrittäminen sivulle
UI_Outfield	Tekstin määrittäminen sivulle
UI_Infield	Vetovalikon määrittäminen sivulle
UI_Separator	Erotinlaatikon määrittäminen

## 5 ESIMERKKIOBJEKTI: POLYURETAANIALUSTA

### 5.1 Parametrit

Valmiiksi mallinnettu polyuretaanialusta tuodaan ArchiCAD-ohjelmaan stl-muotoisena ja muunnetaan GDL-objektiksi. Tämän jälkeen objektiin muokataan halutut parametrit ja luodaan käyttöliittymä, sekä tarvittaessa muokataan ohjelmien sisältöä. Kuvassa 8 esitetään polyuretaanialusta tuotuna ArchiCAD-muotoon. Polyuretaanialustan osalta vakio-parametrien lisäksi luotiin muutama uusi parametri. Uusien parametrien avulla saadaan luotua vetovalikot, joiden avulla pystytään kutsumaan makroista pultit, eristeet, kannakkeet ja korotusjalat polyuretaanialustan päälle tarvittaessa. Parametrit-ikkunassa jokaiselle vetovalikolle luodaan oma tekstikenttä, joka nimetään. Esimerkiksi kuusiokoloruuvit jaetaan varasto- ja tilauskokoihin. Taulukossa 7 esitetään polyuretaanialustalle luodut uudet parametrit.



KUVA 8. Polyuretaanialusta

TAULUKKO 7. Polyuretaanialusta uudet parametrit

Parametri	Selite
Varastokoot	Kuusiokoloruuvien varastokoot
Tilauuskoot	Kuusikoloruuvien tilauuskoot
Kannake	Kannakkeen (ja eristeen) tyyppi
Korotusjalka	Korotusjalan koot
Korko	Kannakkeen ja korotusjalan korko alustaan nähden

## 5.2 Esiohjelma

Esiohjelmassa määritetään muuttujat luoduille parametreille. Tämä tehdään hyödyntämällä ”DIM”-komentoa. Komennon avulla luodaan matriiseja ohjelmaan, mutta sitä voi hyödyntää myös arvolistojen luomiseen. Komennon jälkeen syötetään muuttujan nimi ja hakasulkeiden sisälle, kuinka monta riviä matriisiin halutaan. Arvolistaa varten eri vaihtoehtot saadaan syöttämällä ensin muuttujan nimi ja hakasulkeiden sisälle rivinumero. Yhtäsuuruusmerkin jälkeen heittomerkkien sisälle kirjoitetaan haluttu nimi arvolistan muuttujalle. Tämä toistetaan jokaiselle matriisin riville. Polyuretaanialusta on vakiokokoinen, eikä käyttäjällä ole tarvetta muokata sen kokoa. ”LOCK”-komennolla saadaan lukittua parametrit A, B ja ZZYZX. Tämä estää käyttäjää vahingossakaan muokkaamasta objektin kokoa. Alla on esitetty komentosarja, jonka avulla saadaan määritettyä kuusiokoloruuvien varastokoot-parametrin muuttujat. Liitteessä 1 esitetään kokonaisuudessaan lisätyt arvolistaohjelman komentosarjat.

```

DIM stVarastokoot[5]
stVarastokoot[1] = ""
stVarastokoot[2] = "M12x50/50"
stVarastokoot[3] = "M12x70/70"
stVarastokoot[4] = "M12x80/80"
stVarastokoot[5] = "M12x100/100"

```

### 5.3 3D-ohjelma

3D-ohjelmassa määritetään makrojen hakeminen objektkirjastosta. Kutsuminen tapahtuu ”call”-komennolla. Tietty makro halutaan kutsua kuitenkin vain, jos se on valittuna parametrissa. Tämä tehdään ”ff”-komennon avulla. Tämä komento muodostuu ehtolauseesta, jonka toteutuessa tehdään tietty asia. Tässä tapauksessa halutaan, että ArchiCAD kutsuu parametria vastaavan makro-objektin kirjastosta ja sijoittaa sen polyuretaanialustan päälle. ”Addz”-komennolla saadaan siirrettyä paikallista koordinaatistoa z-suunnassa tarvittavan määrän, jotta objekti on oikeassa kohdassa polyuretaanialustaan nähden. ”Del”-komennolla saadaan paikallinen origo palaamaan aina globaaliin origoon. ”endif”-komento päättää aina ”if”-lausekkeen. Alla on esitetty komentosarja, jonka avulla kutsutaan M12x50/50-kuusiokoloruuvi polyuretaanialusta päälle.

```
IF Varastokoot = "M12x50/50" THEN
    ADDZ 0.002
    CALL "Pultti M12x50"
    DEL 1
ENDIF
```

3D-ohjelmassa määritetään myös ehtolauseke, jonka avulla objektissa ei ole mahdollista kutsua samanaikaisesti kahta kuusiokoloruuvia päällekkäin objektiin. Tämä toteutetaan ”if - else” -komentojen avulla. ”If”-komennon jälkeen syötetään haluttu toiminta. Tässä tapauksessa molemmat valikot halutaan pitää avoimena, mikäli kuusiokoloruuveja ei ole objektissa. ”Else”-komennon jälkeen kirjoitetaan komennot, joiden halutaan toteutuvan, jos edellä määritetty ”if”-lause ei toteudu. Tässä tapauksessa toinen kuusiokoloruuvien valikoista halutaan lukita, jos toisessa on arvona jokin kuusiokoloruuvikoko. Tämä tehdään ”lock”-komennolla, jonka toiminta selitettiin esiohjelma-kappaleessa. Liitteessä 1 esitetään kokonaisuudessaan lisätyt 3D-ohjelman komentosarjat.

### 5.4 Arvolistaohjelma

Arvolistaohjelmassa määritetään arvot parametreille. Tässä tapauksessa määritettäviä parametreja ovat kuusiokoloruuvien, eristeiden, kannakkeiden ja korotusjalkojen valikot. Lisäksi eristeelle ja korotusjalalle määritetään korkovalikko, jonka avulla saadaan säädettyä näiden objektien korkoa kuusiokoloruuviin nähden. Parametrien arvolista luodaan

”Values”-komennon avulla. Komennon jälkeen kirjoitetaan määritettävän parametrin nimi lainausmerkkien sisään. Tämän jälkeen kirjoitetaan halutut muuttujat, jotka erotellaan toisistaan pilkulla. Esiohjelmassa tehtiin jo valmiiksi matriisi, joka sisältää halutut muuttujien nimet, joten arvosta saadaan, kun ”VALUES”-komennon jälkeen kirjoitetaan parametrin nimi sekä matriisin nimi. Valittaessa kannaketta, eristettä ja korotusjalkaa, on määrättävä, että kyseisten objektien korkoasema on polyuretaanialustan yläpinnan ja kuusiokoloruuvien kannan alapinnan välillä. Tämä onnistuu ”RANGE”-komennon avulla. ”Values”-komentoa käytetään kuten edellä ja ”Range”-komennon jälkeen syötetään raja-arvot parametrille. Hakasulje vastaa ArchiCAD-ohjelmassa  $\geq/\leq$ -merkkejä ja kaarisulje  $>/<$ -merkkejä. Alla on esitetty komentosarja, jonka avulla määritetään kuusiokoloruuvien varastokoot-parametrin arvosta. Liitteessä 1 esitetään kokonaisuudessaan lisätyt arvostaohjelman komentosarjat.

VALUES "varastokoot" stVarastokoot
------------------------------------

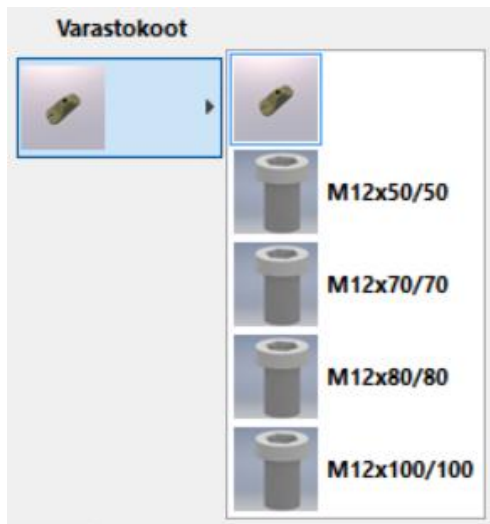
## 5.5 Käyttöliittymäohjelma

Käyttöliittymässä määritellään objektille käyttäjästävällinen käyttöliittymä, jonka avulla voidaan hallinnoida objektin parametreja. Polyuretaanialustan osalta käyttöliittymään sisällytetään myös vetovalikot, joilla saadaan haettua makro-objektit polyuretaanialustaan. Polyuretaanialustan käyttöliittymään tehdään kaksi sivua. Ensimmäiselle sivulle tulevat kuusiokoloruuvien, kannakkeiden ja korotusjalkojen vetovalikot. Toiselle sivulle tehdään objektin leikkausnäkyvän määrittämiseen tarvittavat vetovalikot. Käyttöliittymä koostuu kahdesta osasta; osa sivujen tiedoista määritetään näkymään vain kyseisellä sivulla ja osa painikkeista ja kuvista näkyy joka sivulla. Tämä rakenne tehdään ”GOSUB”-komennon avulla. Kyseisen komennon jälkeen ArchiCAD hyppää muiden rivien ylitse aina ”SUB”-komentoon asti. Lisäksi komennolle määritetään numeroarvo, jonka avulla vastaavia komentoja voidaan tehdä useita. ArchiCAD jatkaa rivien lukemista aina, kunnes komentosarjaan kirjoitetaan ”RETURN”-komento. Tämän jälkeen ArchiCAD palaa ”GOSUB”-komennon jälkeiselle riville ja jatkaa normaalisti ohjelman lukemista.

”GOSUB”-komennolla määritetään käyttöliittymän sellaiset osat, joiden halutaan tulevan useammalle kuin yhdelle sivulle. Täten riittää, että uuden sivun alkuun kirjoittaa komennon, jolla ohjelma käy lukemassa alikomennon komentosarjat, eikä koko komentosarjaa

tarvitse kirjoittaa jokaiselle sivulle uudestaan. Polyuretaanialustan osalta jokaiselle sivulle tehdään eteen- ja taaksepäin -painikkeet, valmistajien kuvat sekä painikkeet valmistajien sivuille. Eteen- ja taaksepäin -painikkeet tehdään yhdistämällä kaksi komentosarjaa. ”UI\_BUTTON”-komennolla saadaan tehtyä painike ja ”UI\_NEXT” tai ”UI\_PREV”-komennolla käyttöliittymä vaihtaa sivua. Näiden jälkeen syötetään lainausmerkkeihin teksti, jonka painikkeessa halutaan näyttää. Seuraavaksi syötetään x- ja y-koordinaatit painikkeelle sekä painikkeen leveys ja korkeus. Vastaavalla tavalla toteutetaan painikkeet internetsivuille. ”UI\_BUTTON”-komennon jälkeen kirjoitetaan ”UI\_LINK”-komento, koordinaatit, painikkeen koko ja viimeiseksi lainausmerkkeihin www-osoite, jonne halutaan päästä. Kuvat käyttöliittymään toteutetaan ”UI\_PICT”-komennolla. Komennon jälkeen syötetään kuvatiedoston nimi, x- ja y-koordinaatit, sekä kuvan haluttu leveys ja korkeus. Kuvatiedoston tulee sijaita jossakin objektikirjaston osassa, jotta kuva saadaan näkyviin objektin käyttöliittymässä.

Sivunvalinta tehdään ”UI\_PAGE”-komennolla, jonka jälkeen kirjoitetaan sivunumero. Vetovalikoiden otsikot tehdään ”UI\_OUTFIELD”-komennolla. Otsikkoteksti kirjoitetaan lainausmerkkeihin komennon perään. Lopuksi syötetään koordinaatit ja kokotiedot. Tekstin tyyliä muokataan ”UI\_STYLE”-komennolla, jonka jälkeen määritetään fonttikoko (0-2) ja tyyli (0-4). Vetovalikot toteutetaan ”UI\_INFIELD”-komennolla. Komennon jälkeen lainausmerkkeihin syötetään haluttu parametri vetovalikkoon. Tämän jälkeen määritetään sijainti ja valikon koko. Seuraavaksi määritetään valikon tyyppi (1-8) ja valikon oletuskuva. Tämän jälkeen määritetään valikon kuvien ja rivien lukumäärä, ja solun sekä esikatselukuvien koko. Vetovalikon vaihtoehtoista määritetään lopuksi kuva ja seliteteksti. Kuvassa 9 esitetään komentosarja, jolla saadaan käyttöliittymään kuusiokolo-ruuvien varastokokojen valintaikkuna ja sen lopputulos käyttöliittymässä. Liitteessä 1 esitetään kirjoitettu käyttöliittymäohjelma kokonaisuudessaan.



UI\_STYLE 2,1

UI\_OUTFIELD "Varastokoot",60,30,100,20

UI\_INFIELD "Varastokoot", 40,50,100,50,  
 2,0,5,5,  
 300,300,40,40,  
 "Polyuretaanialusta", " ",  
 "pultti", "M12x50/50",  
 "pultti", "M12x70/70",  
 "pultti", "M12x80/80",  
 "pultti", "M12x100/100"

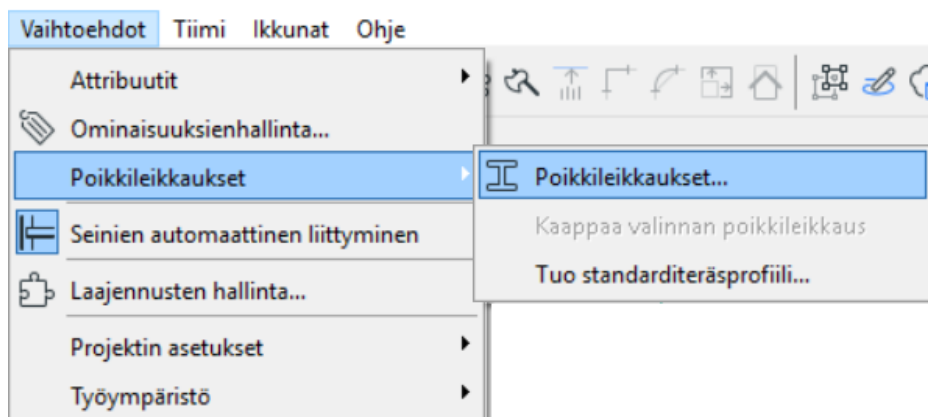
KUVA 9. Käyttöliittymä

## 6 ATTRIBUUTTITIEDOSTOT JA MÄÄRÄLASKENTAPOHJAT

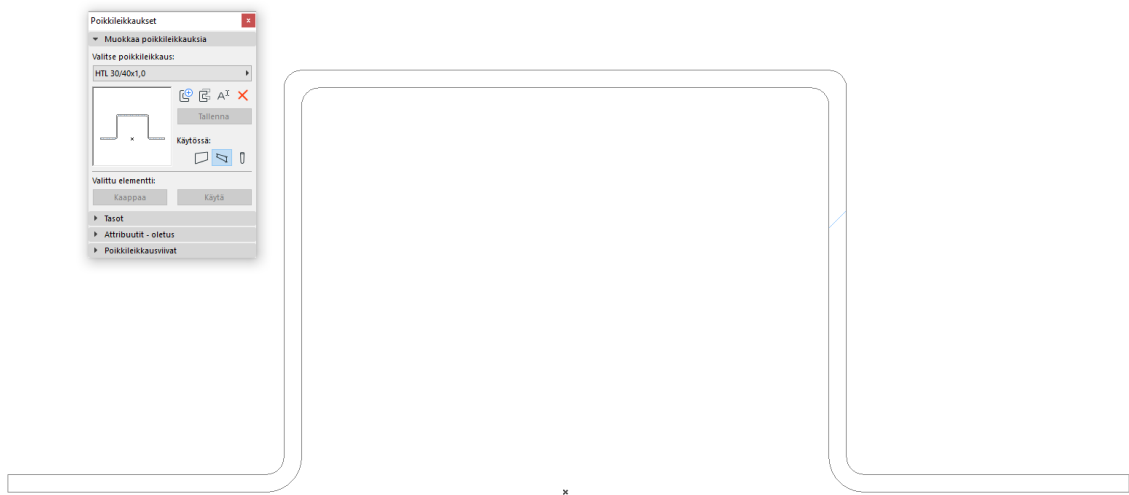
### 6.1 Poikkileikkaukset

Debel-kerroslattiajärjestelmään kuuluu osana erilaisia ja erikokoisia profiileita, jotka toimivat kannattajina kipsilevyllä. Nämä voidaan mallintaa objekteina. Tämä johtaisi kuitenkin kohtuuttoman suureen objektikirjastoon, koska eri profiileita on kymmeniä. Profiilit voitaisiin toki jakaa omaksi kirjastopaketiiksi, mutta järkevämpää on mallintaa profiilit poikkileikkaus-työkalun avulla. Täten mallista tulee myös kevyempi. Profiilien osalta ei ole tarpeellista tehdä käyttöliittymää, sillä ainoa muuttuva attribuutti on pituus, jota on helppo muokata sijoitettaessa profiileita projektiin. ArchiCAD-ohjelmassa tehdyt poikkileikkaukset ja rakennetyypit tallennetaan attribuuttitiedostoina, jotka tunnistaa .aat-tiedostopäätteestä.

Poikkileikkauksien teko tapahtuu valikosta löytyvän poikkileikkaukset-työkalun avulla. Tämän jälkeen piirretään halutun profiilin leikkausprofiili. Täyte-työkalun avulla profiilille valitaan rakennusmateriaali. Tärkeää on tässä kohtaa miettiä profiilin sijainti paikalliseen origoon nähden. Tämä vaikuttaa poikkileikkauksen sijoitukseen malliin sijoitettaessa. Esimerkiksi 20 mm origon yläpuolelle piirretty poikkileikkaus on myös malliin piirrettäessä aina 20 mm profiilin sijoitustason yläpuolella. Poikkileikkausta luodessa pääte-tään työkalut, joiden avulla se on mahdollista sijoittaa malliin. Vaihtoehtoina ovat seinä, palkki ja pilari. Kuvassa 10 esitetään valikkopolku poikkileikkauksen luomiseen ja kuvassa 11 esimerkkipoikkileikkauksena HTL 30/40x1,0 -profiili. Profiili on käytössä palkki-työkalulla ja se on piirretty teräs-täytteellä.



KUVA 10. Valikkopolku poikkileikkauksien luomiseen

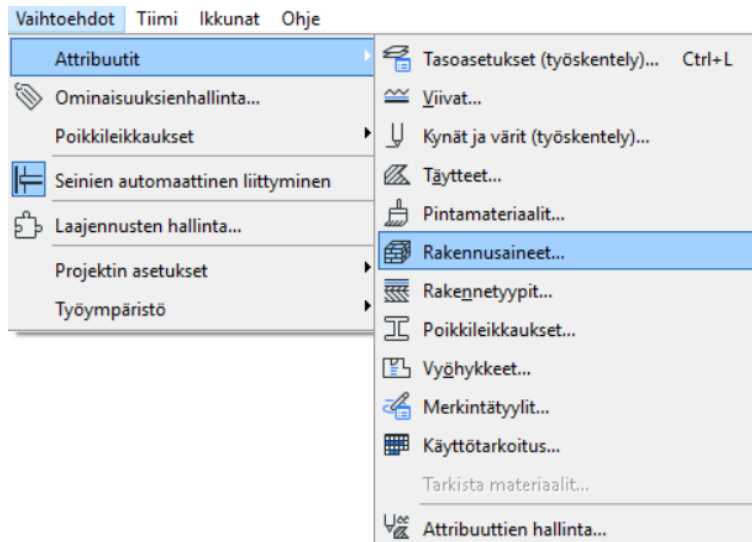


KUVA 11. HTL 30/40x1,0 -poikkileikkausprofiili

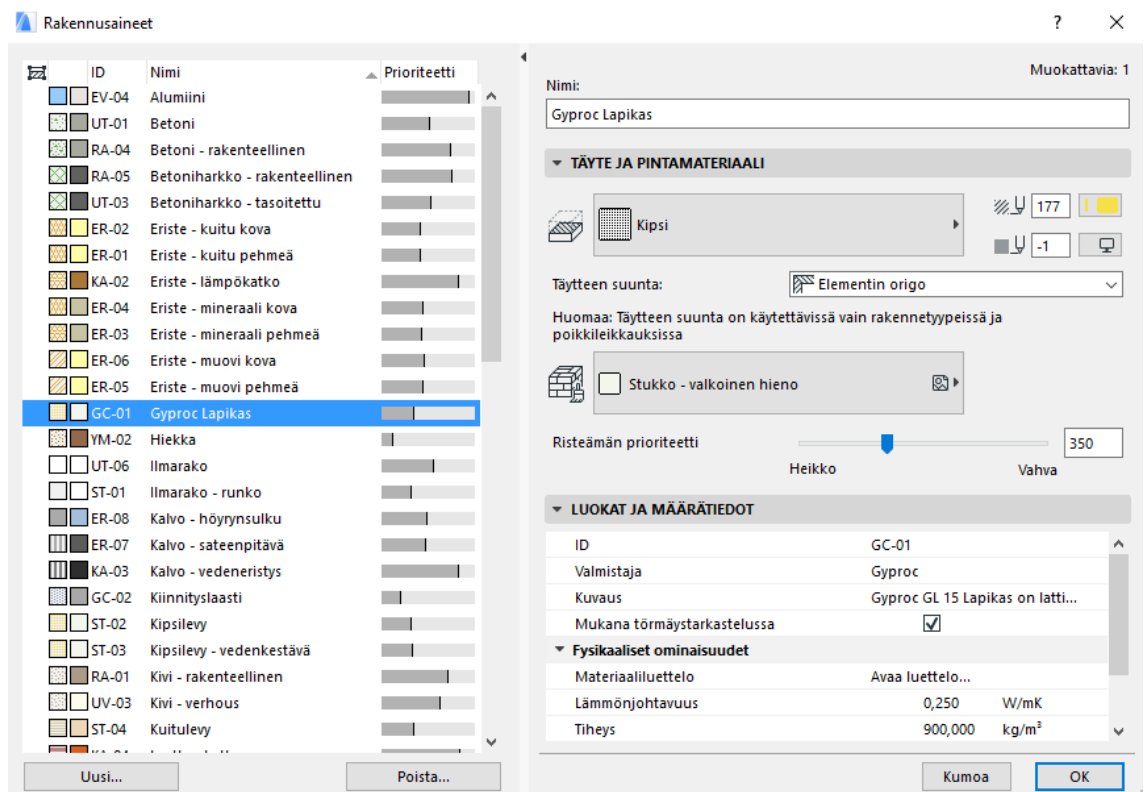
## 6.2 Rakennusaineet

Rakennetyypit koostuvat rakennusaineista. ArchiCAD-ohjelmassa on yleisimmät rakennusaineet, kuten kipsilevy ja betoni, valmiina. Rakennusaineita pystyy luomaan myös itse rajattomasti lisää. Vähänkään erilaiset rakennusaineet kannattaa erotella omiksi, koska määrälaskentavaiheessa ne saadaan eroteltua omiksi kokonaisuuksiksi. Esimerkiksi normaalille ja erikoiskovalle kipsilevyille saadaan eroteltua omat määrälaskentatulokset.

Rakennusainetta luodessa sille määritetään täytemateriaali sekä pintamateriaali. Rakennusaineen prioriteetti määrittää, kuinka helposti päällekkäiset rakennusaineet leikkaavat toisen rakennusaineen läpi. Lisäksi rakennusaineelle voidaan syöttää tietoja valmistajasta ja materiaaliominaisuuksista. Tässä työssä luotiin uudet rakennusaineet Gyproc Lapikkaalle ja sen kiinnityslaastille. Kuvassa 12 esitetään valikkopolku rakennusaineen luomiseen ja kuvassa 13 esitetään luotu Gyproc Lapikas -rakennusaine.



KUVA 12. Valikkopolku rakennusaineiden luomiseen

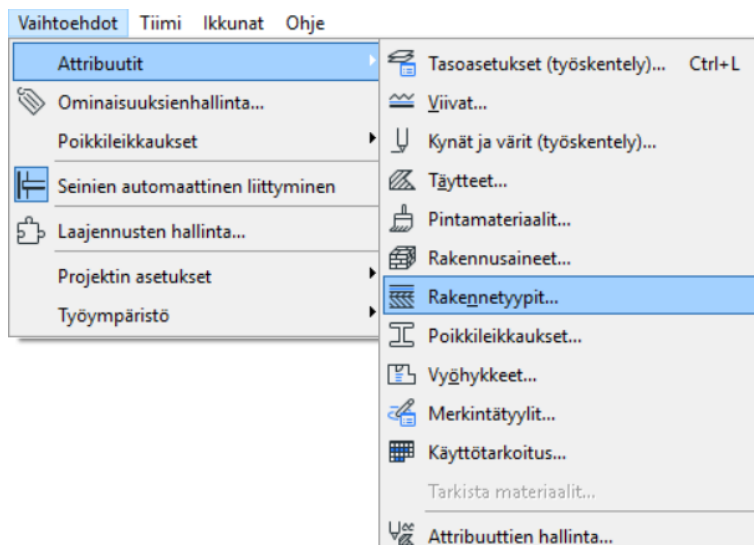


KUVA 13. Gyproc Lapikas -rakennusaine

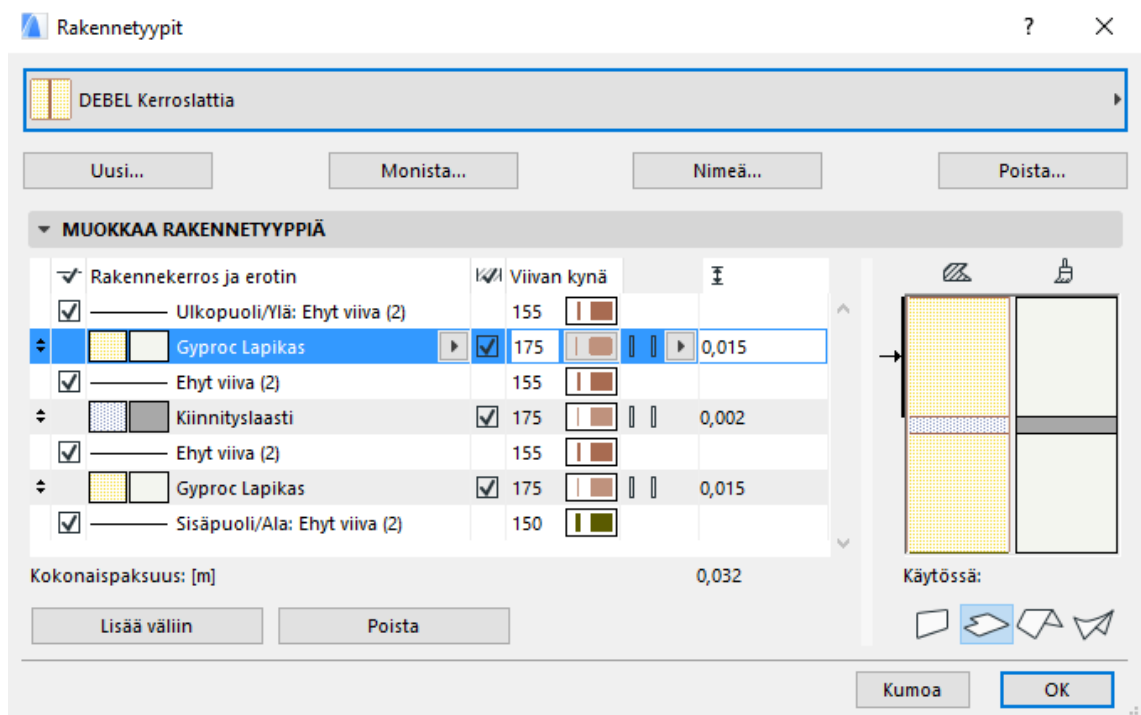
### 6.3 Rakennetyypit

Rakennetyyppien avulla voidaan mallintaa eri rakennekokonaisuuksia nopeasti ja vaivattomasti ilman että jokainen rakennekerros täytyy piirtää yksitellen. ArchiCAD-rakennetyypin luomisen jälkeen se vain piirretään valitulla työkalulla malliin. Esimerkiksi koko yläpohjarakenne saadaan piirrettyä laatta-työkalulla malliin nopeasti, kunhan se on ensin

tehty rakennetyypiksi. Rakennetyyppiä luodessa haluttu rakennusaine valitaan vetovalikosta ja sille määritetään paksuus. Rakennusaineelle määritetään kynän väri, jolla se näkyy leikkauksissa, sekä reunat joille viivat piirretään. Tämä toistetaan jokaisen rakennetyypin kuuluvan rakenneosan osalta. Rakennetyyppiä luodessa valitaan työkalut, joiden avulla se on mahdollista piirtää. Vaihtoehtoina ovat seinä, laatta, katto ja kuori. Kuvassa 14 esitetään valikkopolku rakennetyypien luomiseen ja kuvassa 15 esitetään Debel-kerroslattian rakennetyyppi. Debel-kerroslattian rakennetyyppi koostuu kahdesta päällekkäisestä Gyproc Lapikas -kipsilevystä, jotka kiinnitetään toisiinsa liimalla tai laastilla.



KUVA 14. Valikkopolku rakennetyypien luomiseen

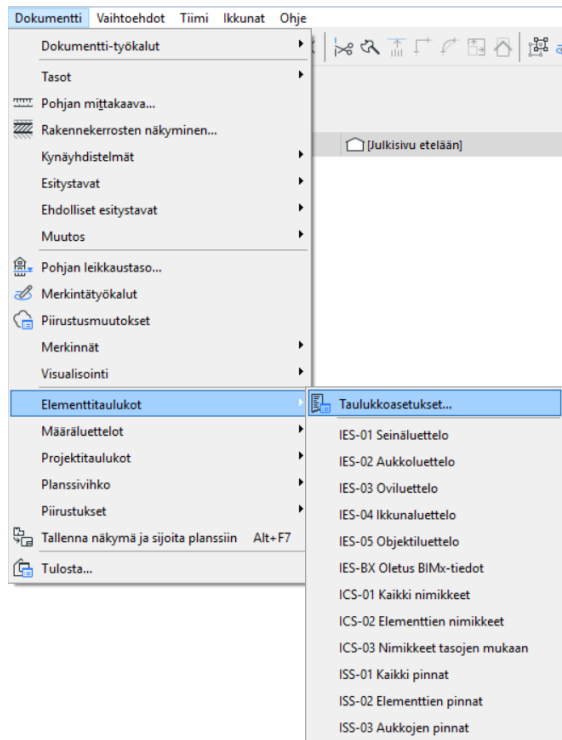


KUVA 15. Debel-kerroslattian rakennetyyppi

## 6.4 Määrälaskentapohjat

Määrälaskennan avulla mallista saadaan nopeasti tuotua tuloksia, joiden avulla tuotteiden tilaus on helppoa. Kipsilevyille saadaan laskettua neliömäärät, objekteille kappalemäärät ja profiileille tarkat metrimäärät.

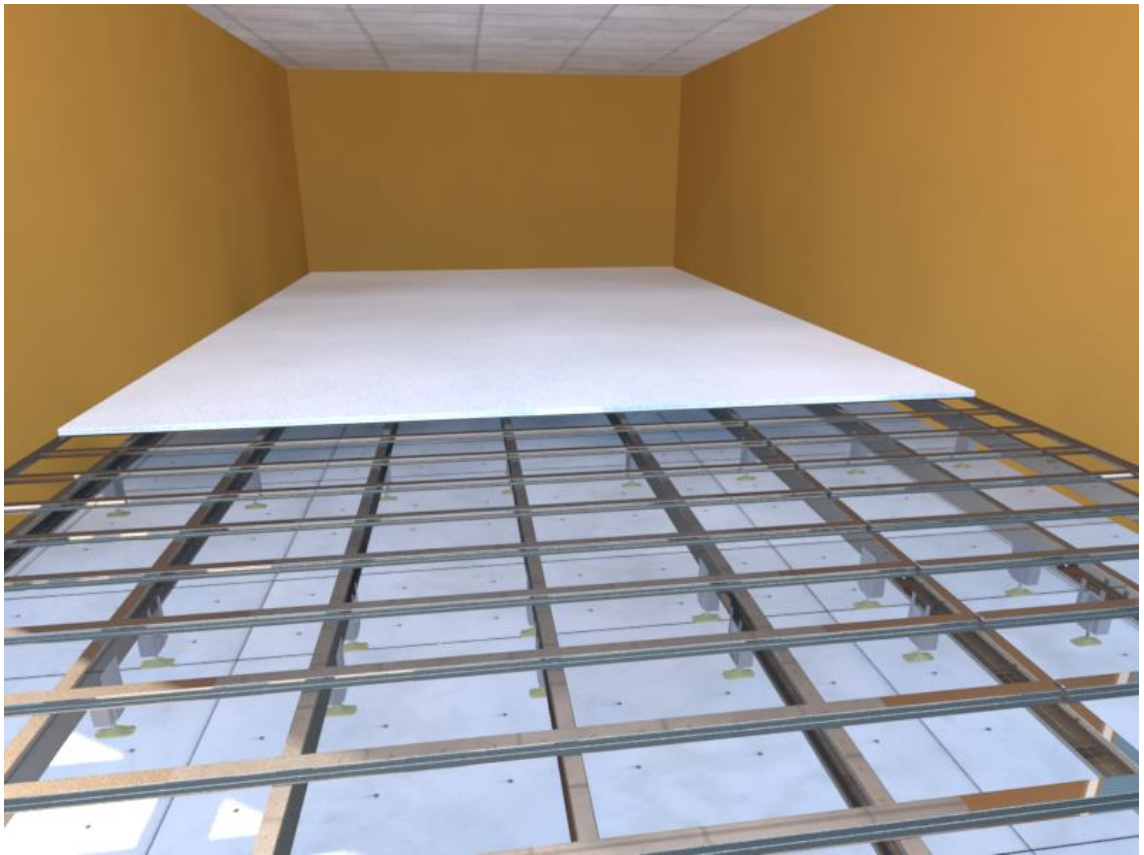
Tässä työssä määrälaskentapohjat tehtiin taulukoiden elementtitaulukoiden avulla. Valmis taulukko saadaan tallennettua eri tiedostomuotoihin, esimerkiksi .xls-tiedostoksi, jossa sitä on helppo muokata jatkossa. Uutta taulukkoa luodessa valitaan ensin, taulukoitaanko rakenne-elementtejä, nimikkeitä vaiko pintoja. Taulukolle määritetään ID-tieto, jolla taulukot järjestellään ohjelmassa. Attribuuttien avulla määritetään ehdot, joilla halutut tiedot saadaan haettua mallista ja tarpeettomat tiedot saadaan suodatettua pois. Esimerkiksi Debel-objektiluettelossa määritetään, että ArchiCAD laskee mallista vain objektit, joiden nimet kuuluvat Debel-objektikirjastoon. Laskettaviin tietoihin syötetään tiedot, jotka halutaan tietää edellä määritetyistä objekteista. Esimerkiksi objektin nimet ja niiden kappalemäärät. Tietojen oikealla puolella olevilla napeilla määritetään järjestyksiä ja laskettavia tietoja. Nuolella määritellään joko laskeva tai nouseva järjestys valitulle tiedolle, esimerkiksi aakkosjärjestys. Mikäli nuolia on useampi, taulukko järjestää seuraavan sarakkeen omassa järjestyksessään. Esimerkiksi eripituiset profiilit voidaan järjestellä ensin aakkosjärjestykseen ja tämän jälkeen pituuden mukaan kasvavaan järjestykseen. Summa-napilla määritellään yhteenlaskettavia tietoja. Pelkällä  $\Sigma$ -merkillä taulukossa lasketaan kyseisen sarakeosien summa.  $\Sigma_1$ -merkillä taulukkoon lasketaan kokonaismäärä kyseisestä sarakkeesta. Lippu-merkillä saadaan välisummia identtisille elementeille. Esimerkiksi profiililtaan samanlaisten, mutta eripituisten profiilien kappalemäärät saadaan tällä tavalla laskettua. Lippu-merkittynä voi olla vain yksi sarake kerrallaan. Kuvassa 16 esitetään valikkopolku elementtitaulukoiden luomiseen. Liitteessä 2 on kuvitteellisen esimerkkikohteen määrälaskentataulukoiden tulokset.



KUVA 16. Valikkopolku määrälaskentataulukon luomiseen

## 6.5 Esimerkkikohde

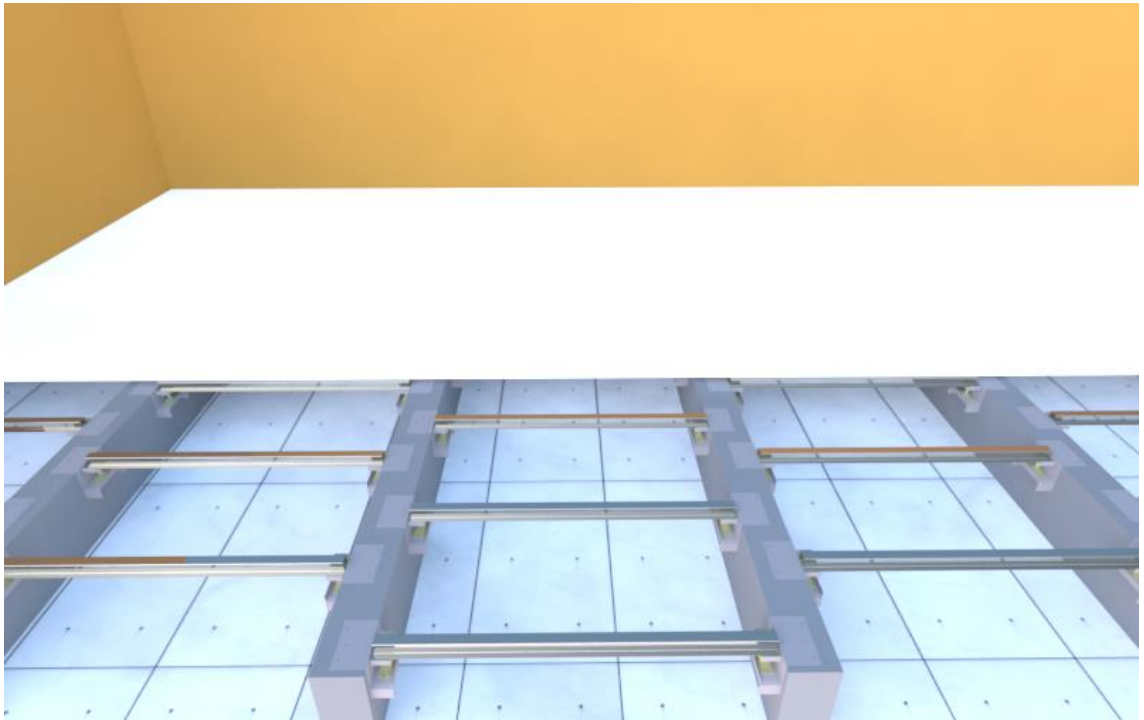
Luodaan objektien ja profiilien avulla kuvitteellinen esimerkkikohde suunnittelua varten. Kohteessa on kolme kerrosta, joissa on 50 neliötä lattiapinta-alaa. Ensimmäisessä kerroksessa (kuva 17) on Debel-kerroslattian perusrakenne, jossa on polyuretaanialustat k1200-jaolla. Kantavana rakenteena on KC-profiili k600-jaolla ja toisiokannattajana SKH-profiili k300 jaolla. Toisessa kerroksessa (kuva 18) on madallettu rakenne. Tässä polyuretaanialustat ovat k900-jaolla. Kantavana rakenteena toimivat HTL-profiilit k400-jaolla. Kolmannessa kerroksessa (kuva 19) Debel-kerroslattia on tehty alalaattapalkiston päälle. Tässä polyuretaanialusta on Z-kannattajan päällä, joka on kiinnitetty alalaattapalkistoon. Kantavana rakenteena on HTL-profiili k600-jaolla. Liitteessä 2 esitetään tästä kuvitteellisesta kohteesta tehdyt määrälaskentatulokset.



KUVA 17. Ensimmäisen kerroksen rakenne



KUVA 18. Toisen kerroksen rakenne



KUVA 19. Kolmannen kerroksen rakenne

## 7 POHDINTA

GDL-objektien suurin haaste on ohjelmointikielen rajallinen levinneisyys. GDL on irtaantunut BASIC-ohjelmointikielestä, eikä GDL-kieltä opeteta koulutuslaitoksissa. Koulutuksia järjestää Suomessa vain M.A.D. Oy eikä ohjetekstejä ole montakaan. GDL-objekteissa on runsaasti älykkäitä ominaisuuksia ja niissä olisi potentiaalia kehitykselle jatkuvasti rakentamisessa lisääntyvässä tietomallinnuksessa. Älykkäisiin objekteihin pystytään sitomaan tietoa, jota pystytään hyödyntämään rakennushankkeen eri vaiheissa. Graphisoftin kannattaisi panostaa GDL-kielen levittämiseen ja koulutukseen. GDL-editoriin kannattaisi myös kehittää kunnollinen komentosarjojen tarkistustyökalu, jolla virheellinen komentosarja löytyisi helpommin.

Objektikirjaston osalta suunnittelua etukäteen ei voi korostaa tarpeeksi. Suositeltavaa olisi, että objektien vaatimuksista ja toiveista käytäisiin tilaajan kanssa palaveri. Ohjelmoijan kannattaa tehdä esimerkiksi vuokaavio, jossa hän pohtii objektin rakennetta ja miten eri ominaisuudet kannattaa sijoittaa eri ohjelmiin. Tämän työn osalta objektien käyttöliittymää muokattiin työn edetessä tilaajien toiveiden mukaisesti. Nämä muutokset aiheuttivat muokkauksia objektin rakenteeseen.

Opinnäytetyössä saatiin tehtyä tilaajalle valmis kirjastopaketti, jonka avulla tulevien kohteiden mallintaminen onnistuu nopeasti. Määrälaskentapohjien avulla kohteeseen tarvittavien komponenttien, profiilien ja levyjen laskenta on vaivatonta. Nämä tulokset saadaan suoraan Excel-taulukkomuotoon, josta tuotetilaus ja tietojen jatkojalostaminen ovat yksinkertaista. Tämän työn pohjalta tilaajan on mahdollista alkaa kehittää vastaavanlaisia kirjastopaketteja myös muista heidän rakennusjärjestelmistään.

## LÄHTEET

Aulis Lundell Oy. 2016. ProfAL korotuslattia. Tuoteohje. Luettu 27.3.2017.  
[http://www.aulislundell.fi/upload/ProfAL\\_lattia\\_tuoteohje\\_2016\\_web.pdf](http://www.aulislundell.fi/upload/ProfAL_lattia_tuoteohje_2016_web.pdf)

Chopra, A. 2011. Google Skethup 8 for Dummies. Indianapolis: Wiley Publishing, Inc.

Hietikko, E. 2007. Autodesk Inventor. 1 Painos. Helsinki: Readme.fi.

Hyttinen, A., Rissanen, J., Lindroos, P., Kemppainen, H. & Nyholm, M. 2016. Gyproc Käsikirja. Kirkkonummi: Saint Gobain Rakennustuotteet / Gyproc.

MacKenzie, S & Rendek Adam. 2015. ArchiCAD 19 – The Definitive Guide. 1. Painos. Birmingham: Packt Publishing Limited.

Nicholson-Cole, D. GDL Cookbook 4. Luettu 27.3.2017.  
<http://www.btsquarepeg.com/2008/03/11/download-gdl-cookbook-4-pdf/>

Saint-Gobain Rakennustuotteet Oy. 2015. Debel-kerroslattiajärjestelmä. Luettu 27.3.2017. <http://www.gyproc.fi/ratkaisut/lattiat/debel-kerroslattiajarjestelma>

## LIITTEET

### Liite 1. Polyuretaanialustan ohjelmat

1 (7)

#### ESIOHJELMA

```

! =====
!! Pulttien kokovalikko
! =====

DIM stVarastokoot[5]
stVarastokoot[1] = "
stVarastokoot[2] = 'M12x50/50'
stVarastokoot[3] = 'M12x70/70'
stVarastokoot[4] = 'M12x80/80'
stVarastokoot[5] = 'M12x100/100'

DIM stTilaukoot[5]
stTilaukoot[1] = "
stTilaukoot[2] = 'M12x40/40'
stTilaukoot[3] = 'M12x60/60'
stTilaukoot[4] = 'M12x90/90'
stTilaukoot[5] = 'M12x120/120'

IF varastokoot = stVarastokoot[1] THEN
    ELSE
        LOCK "tilaukoot"
ENDIF

IF tilaukoot = stTilaukoot[1] THEN
    ELSE
        LOCK "varastokoot"
ENDIF

! =====
!! Kannakkeiden valikko
! =====

DIM stKannake[3]
stKannake[1] = "
stKannake[2] = 'HTL'
stKannake[3] = 'KC'

IF Kannake = stKannake[3] THEN
    ELSE
        LOCK "Korotusjalka"
ENDIF

```

```

! =====
!! Korotusjalkojen valikko
! =====

dim stKorotusjalka[4]
stKorotusjalka[1] = "
stKorotusjalka[2] = '155'
stKorotusjalka[3] = '450'
stKorotusjalka[4] = '800'

IF Korotusjalka = stKorotusjalka[1] THEN

        ELSE

                LOCK "Kannake"

ENDIF

LOCK "A","B","ZZYZX"

```

### 3D-OHJELMA

```

! =====
!Kannakkeiden hakeminen makroista
! =====

IF Kannake = "HTL" THEN

        ADDZ korko

                CALL "HTL-kannake"

        ADDZ 0.003

                CALL "HTL-eriste"

                        DEL 2

        ENDIF

IF Kannake = "KC" THEN

        ADDZ korko

                CALL "KC-kannake"

        ADDZ 0.003

                CALL "KC-eriste"

                        DEL 2

        ENDIF

! =====
!Korotusjalan hakeminen makroista
! =====

IF Korotusjalka = "155" then

        ADDZ Korko+0.013

                CALL "Korotusjalka 155"

                        DEL 1

        ENDIF

IF Korotusjalka = "450" THEN

        ADDZ Korko+0.013

                CALL "Korotusjalka 450"

                        DEL 1

```

```

ENDIF
IF Korotusjalka = "800" THEN
    ADDZ Koroko+0.013
        CALL "Korotusjalka 800"
            DEL 1
    ENDIF
ENDIF
IF Korotusjalka = "" THEN
    ENDIF
! =====
!Pulttien hakeminen makroista
! =====
    IF Varastokoot = "M12x50/50" THEN
        ADDZ 0.002
            CALL "Pultti M12x50"
                DEL 1
    ENDIF
    IF Varastokoot = "M12x70/70" THEN
        ADDZ 0.002
            CALL "Pultti M12x70"
                DEL 1
    ENDIF
    IF Varastokoot = "M12x80/80" THEN
        ADDZ 0.002
            CALL "Pultti M12x80"
                DEL 1
    ENDIF
    IF Varastokoot = "M12x100/100" THEN
        ADDZ 0.002
            CALL "Pultti M12x100"
                ADDZ 0.002
                DEL 1
    ENDIF
    IF Varastokoot = " " THEN
    ENDIF

    IF Tilaskoot = " " THEN
    ENDIF
    IF Tilaskoot = "M12x40/40" THEN
        ADDZ 0.002
            CALL "Pultti M12x40"
                DEL 1
    ENDIF
    IF Tilaskoot = "M12x60/60" THEN
        ADDZ 0.002
            CALL "Pultti M12x60"
                DEL 1
    ENDIF
    IF Tilaskoot = "M12x90/90" THEN
        ADDZ 0.002
            CALL "Pultti M12x90"

```

```

DEL 1
ENDIF
IF Tilauskoot = "M12x120/120" THEN
    ADDZ 0.002

                                CALL "Pultti M12x120"
                                DEL 1
ENDIF

```

## ARVOLISTA OHJELMA

```

! =====
! Debel valikko
! =====
VALUES "varastokoot" stVarastokoot
VALUES "tilauskoot" stTilauskoot
VALUES "Korotusjalka" stKorotusjalka
VALUES "Kannake" stKannake

    IF varastokoot = "M12x50/50" THEN
        VALUES "Korko" RANGE [0.017, 0.034]
        ENDIF

    IF varastokoot = "M12x70/70" THEN
        VALUES "Korko" RANGE [0.017, 0.054]
        ENDIF

    IF varastokoot = "M12x80/80" THEN
        VALUES "Korko" RANGE [0.017, 0.064]
        ENDIF

    IF varastokoot = "M12x100/100" THEN
        VALUES "Korko" RANGE [0.017, 0.084]
        ENDIF

    IF tilauskoot = "M12x40/40" THEN
        VALUES "Korko" RANGE [0.017, 0.024]
        ENDIF

    IF tilauskoot = "M12x60/60" THEN
        VALUES "Korko" RANGE [0.017, 0.044]
        ENDIF

    IF tilauskoot = "M12x90/90" THEN
        VALUES "Korko" RANGE [0.017, 0.074]
        ENDIF

    IF tilauskoot = "M12x120/120" THEN
        VALUES "Korko" RANGE [0.017, 0.104]
        ENDIF

    IF varastokoot = "" THEN
        VALUES "Korko" RANGE [0.017,]
        ENDIF

    IF tilauskoot = "" THEN
        VALUES "Korko" RANGE [0.017,]
        ENDIF

```

## KÄYTTÖLITTYMÄOHJELMA

```
UI_DIALOG 'Polyuretaanialustan asetukset'
```

```
!-----
```

```
UI_PAGE 1
```

```
GOSUB 100
```

```
UI_STYLE 2,1
```

```
UI_OUTFIELD "Kuusiokoloruuvien valinta",100,5,150,20
```

```
UI_STYLE 2,1
```

```
UI_OUTFIELD "Varastokoot",60,30,100,20
```

```
UI_OUTFIELD "Tilaukoot",220,30,100,20
```

```
UI_INFIELD "Varastokoot",
                                     40,50,100,50,
                                     2,0,5,5,
                                     300,300,40,40,
                                     "Polyuretaanialusta"," ",
                                     "pultti","M12x50/50",
                                     "pultti","M12x70/70",
                                     "pultti","M12x80/80",
                                     "pultti","M12x100/100"

UI_INFIELD "tilaukoot",
                                     200,50,100,50,
                                     2,0,5,5,
                                     100,100,40,40,
                                     "Polyuretaanialusta"," ",
                                     "pultti","M12x50/50",
                                     "pultti","M12x70/70",
                                     "pultti","M12x80/80",
                                     "pultti","M12x100/100"
```

```
UI_SEPARATOR 0,110,390,110
```

```
UI_STYLE 2,1
```

```
UI_OUTFIELD "Kannake",65,120,150,20
```

```
UI_OUTFIELD "Korotusjalka",220,120,150,20
```

UI\_OUTFIELD "Kannakkeen korko",40,210,150,20

UI\_INFIELD "Kannake", 40,150,100,50,  
 2,0,3,3,  
 300,300,40,40,  
 "Polyuretaanialusta"," ",  
 "HTL-kuva","HTL",  
 "KC-kuva","KC"

UI\_INFIELD "Korotusjalka", 200,150,100,50,  
 2,0,4,4,  
 300,300,40,40,  
 "Polyuretaanialusta"," ",  
 "Korotusjalka-kuva","155",  
 "Korotusjalka-kuva","450",  
 "Korotusjalka-kuva","800"

UI\_INFIELD "Korko", 40,230,70,20

!-----

UI\_PAGE 2

GOSUB 100

UI\_STYLE 2,1

UI\_OUTFIELD "Pohjanäkymä", 5, 0, 140, 20  
 UI\_OUTFIELD "Näytä projektio", 220, 0, 140, 20  
 UI\_OUTFIELD "Symbolin viivakynä", 5, 45, 140, 20  
 UI\_OUTFIELD "Leikkaustäyte", 5, 95, 140, 20  
 UI\_OUTFIELD "Leikkauksen viivakynä", 5, 145, 140, 20  
 UI\_OUTFIELD "Leikkauksen täytteen kynä", 5, 195, 140, 20  
 UI\_OUTFIELD "Leikkausviivat", 220, 45, 150, 20  
 UI\_OUTFIELD "Leikkaustäytteen taustakynä", 220, 95, 150, 20  
 UI\_OUTFIELD "Yläpuolen viivatyyppi", 220, 145, 150, 20  
 UI\_OUTFIELD "Yläpuolen viivakynä", 220, 195, 150, 20

UI\_STYLE 0,0

UI\_INFIELD "gs\_symb\_display\_option", 5, 15, 200, 25  
 UI\_INFIELD "gs\_symb\_show\_projection\_to", 220, 15, 170, 25  
 UI\_INFIELD "gs\_cont\_pen", 5, 65, 140, 25

```
UI_INFIELD "gs_fill_type",           5, 115, 140, 25
UI_INFIELD "gs_fill_pen",           5, 165, 140, 25
UI_INFIELD "gs_back_pen",           5, 210, 140, 25
UI_INFIELD "AC_cut_linetype",       220, 65, 150, 25
UI_INFIELD "AC_uncut_pen",          220, 115, 150, 25
UI_INFIELD "AC_overhead_linetype",  220, 165, 150, 25
UI_INFIELD "AC_overhead_pen",       220, 210, 150, 25
```

END

!-----

100:

UI\_SEPARATOR 390,0,390,450

UI\_BUTTON UI\_PREV, "<--", 400,240, 70,20

UI\_BUTTON UI\_NEXT, "-->", 470,240, 70,20

UI\_PICT "gyproc", 400,0,130,52

UI\_PICT "lundell", 400,110,107,84

UI\_BUTTON UI\_LINK, "Gyproc.fi", 400,60,130,20,0,

"http://www.gyproc.fi"

UI\_BUTTON UI\_LINK, "Debel rakennetyypit", 400,85,130,20,0,

"http://www.gyproc.fi/suunnittelu/rakennekirjasto/debel-kerroslattia"

UI\_BUTTON UI\_LINK, "Aulis Lundell", 400,200,130,20,0,

"http://www.aulislundell.fi/"

RETURN

## Liite 2. Esimerkkikohteen määrälaskennat

Objekttiluettelo					
Objektin nimi	Kappalemäärä	Kannake	Korotusjalka	Tilauuskoot	Varastokoot
Polyuretaanialusta	80	KC	155		M12x80/80
Polyuretaanialusta	144	HTL			M12x50/50
Polyuretaanialusta	139	HTL			M12x80/80
	363				
Z-kannake	136	---	---	---	---
	136				
	499				

Debel-profiilit		
Rakennusmateriaali / Poikkileikkaus	Määrä	Pituus
HTL 35/40x1,5	24	3,30
HTL 35/40x1,5	12	3,13
HTL 35/40x1,5	60	1,14
HTL 35/40x1,5	8	0,57
		189,72 m
J 21/60x1,0	2	4,65
		9,30 m
KC 100x1,5	16	4,00
KC 100x1,5	8	1,73
		77,84 m
SKH_21x1,0	32	3,30
SKH_21x1,0	32	1,41
		150,72 m
		427,58 m

Debel kerros lattia	
Rakennetyyppi / Poikkileikkaus	Laatan yläpinnan bruttoala aukkoineen
DEBEL Kerros lattia	46,31
DEBEL Kerros lattia	46,30
DEBEL Kerros lattia	50,00
3	142,61 m <sup>2</sup>

Gyproc Lapikas	
Nimi	Rakennekerros/-osan pinta-ala
Gyproc Lapikas	285,22
	285,22 m <sup>2</sup>