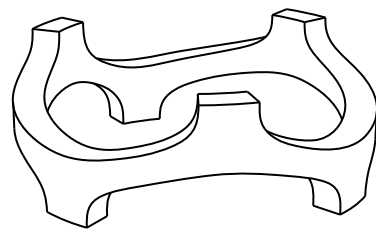


Sattuma-sarja

Parametrein muotoiltuja sormuksia



Lahden ammattikorkeakoulu

Muotoilu- ja taideinstituutti

Muotoilun koulutusohjelma

Taideteollisuuden suuntautumisvaihtoehto

Korumuotoilu

Opinnäytetyö AMK

Kevät 2014

Mikko Vitikka

Ohjaaja: Pekka Koponen

Opponentti: Hanna Silander

Tiivistelmä

Muotoilu on ongelmien ratkointaa. Ratkaisuja voidaan esittää ja kokeilla erilaisten mallien avulla. Parametriset mallit ovat tehokkaita muodon haussa ja eri vaihtoehtojen kokeilussa. Tässä opinnäytetyössä tutkin ja sovellan viime vuosina yleistyneitä 3D-mallinnusohjelmien työkaluja, joilla rakennetaan parametrisia malleja. Hyödynnän taitojani Saarikorpi Design Oy:lle suunnittelemissani sormussarjassa, jossa sormukset ovat selkeästi samaa perhettä, mutta silti muodoltaan uniikkeja. Sormukset on tarkoitettu kihla- ja vihkikäyttöön ja siksi ne luodaan pareittain. Parin sormukset voidaan liittää yhteen, sillä sormuksissa on pinnat, jotka vastaavat toisiinsa.

Abstract

Designing is solving problems. Solutions can be presented and tested with models. Parametric models in particular are effective in exploring and finding forms. During the last decade the parametric tools of 3D modeling softwares have become more common. In my thesis I have studied those tools and applied them in design process of a series of wedding and engagement rings for Saarikorpi Design Ltd. The similarity of the rings is clear but they are not copies of each other. Each ring is unique and has its designated pair that matches its connecting faces.

Sisällysluettelo

Johdanto	6
Parametrit ja muotoilu	6
Parametristen mallien ominaisuudet ja soveltuvuus	8
Historiaa	9
Saarikorpi Design Oy:n toimeksianto	11
Idean kehitys	11
Malli	16
Tuloksia ja kehitysideoita	25
Lähteet	29

Johdanto

Onnellinen vahinko kuvaa hyvin tapaa, jolla päädyin tutustumaan parametriseen muotoiluun. Aihe liittyy tiiviisti tietokoneohjelmiin, joilla tehdään malleja muotoilun avuksi. Tällaisia ohjelmia on ollut yleisesti saatavilla suurin piirtein yhtä kauan kuin tietokoneitakin. Perinteisesti niiden avulla on määritetty staattisia pintoja ja geometriaa, jotka rajaavat muotoja avaruudesta. Viime vuosina yleistyneet ominaisuudet vievät pintaa syvemmälle. Käyttäjä voi määritellä oman säännöstönsä, jonka mukaan muodot syntyvät. Kerron tästä tarkemmin seuraavassa luvussa.

Keskityin opettelemaan Rhinoceros-ohjelman kahden eri lisäosan eli *plug-inin* käyttöä. Pysin samalla olemaan tietoinen muusta tarjonnasta ja seuraava luku käsittelee parametrista muotoilua yleisesti erikoistumatta mihinkään ohjelmaan. Esimerkit on luotu Rhinoceros ja Grasshopper -ohjelmilla, mutta oletan muidenkin ohjelmien pystyvän samaan.

Välillisesti Saarikorpi Design Oy:n ansiosta kiinnostuin koko aiheesta. Siksi tuntui luontevalta soveltaa uusia taitojani suunnittelemaan koruja samaiselle yritykselle. Muotoiluni päätuote on parametrinen malli sormus-sarjasta. Tästä ja onnellisesta vahingosta kerron omassa luvussaan.

Parametrit ja muotoilu

Parametrinen muotoilu on tapa määrittellä muotoja. Parametrit ovat muuttujia, joiden avulla idea muodosta esitetään ja joiden ansiosta muotoilusta tulee muunneltavaa. Muunneltavuus ja säännönmukaisuus ovat yhtä aikaa parametrisen muotoilun vahvuuksia, jotka tulevat esille tuotesarjoja tai muita toistuvia ja hieman toisistaan poikkeavia kohteita suunniteltaessa. Kyse on muotoilun muotoilusta.

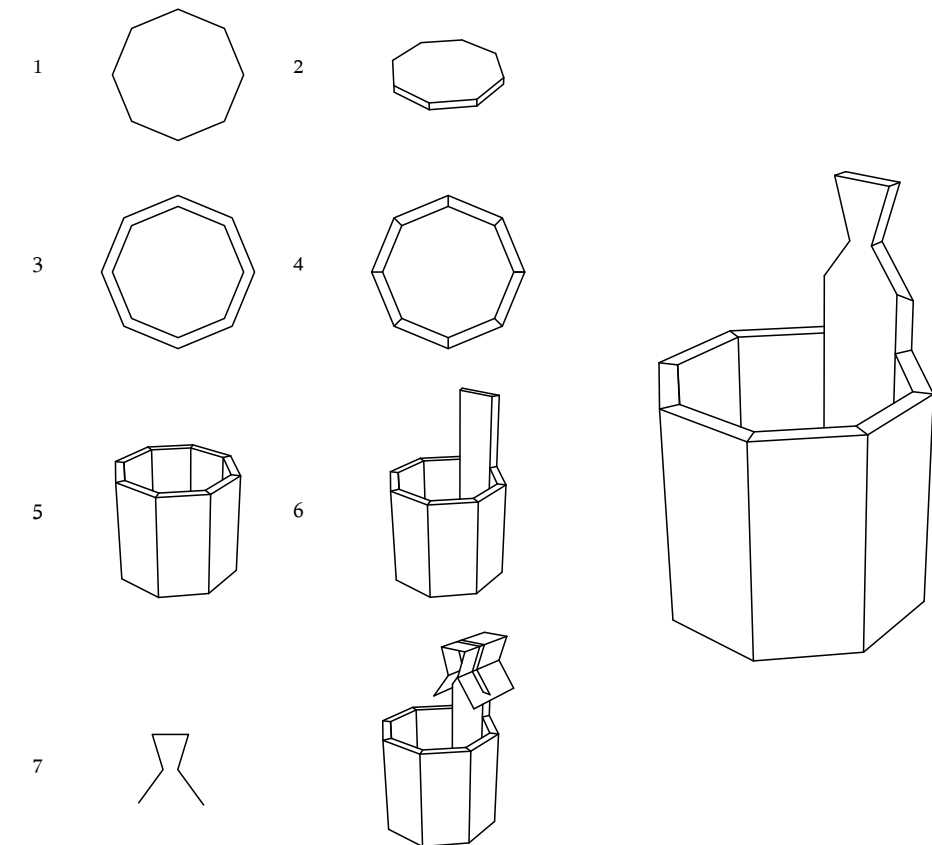
Yksiselitteisimpiä muuttujia ovat luvut, joilla voidaan ilmaista vaikkapa tavoiteltua mittasuhdetta tai lukumäärää. Lukujen käsittely matemaattisesti on yhä helpompaa alati kehittyvän tietotekniikan ansiosta. Vaikka parametrinen muotoilu on mahdollista ilman tietokoneitakin, juuri ne antavat sille uusia mahdollisuuksia. On mielekästä luoda muotoja matemaattisten käyrien avulla kun kone laskee ja piirtää muodot tarkasti muotoilijan puolesta.

Parametrinen muotoilu on muotojen käyttäytymissääntöjen määrittelyä. Perusidean pohjalta kohteelle luodaan erilaisia ominaisuuksia ja määritellään, mitkä niistä ovat säädettävissä ja mitkä pysyvät vakioina. Esimerkiksi tietokoneen käyttöjärjestelmän ikkunat muuttavat kokoaan käyttäjän tahdon mukaan, mutta ulkoasu, kuten painikkeet ja valikkorivi säilyvät ikkunan muodon suhteen omilla paikoillaan. Parametrein voidaan siis muotoilla tyylejä.

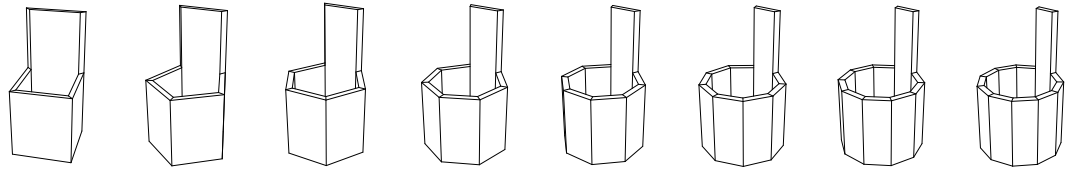
Erilaiset tietokoneohjelmat mahdollistavat kolmiulotteisten mallien luomisen halutun geometrian, eli erilaisten viivojen ja peruskappaleiden, pohjalta. Perinteisesti *parametrittömiä* malleja rakennetaan eli mallinnetaan vaihe vaiheelta luomalla ja muokkaamalla viivoja ja pintoja. Monien työvaiheiden jälkeen alussa määriteltyjä perusominaisuuksia voi olla hankala muuttaa.

Verrattain yksinkertainen esimerkki laudoista koostuvan kiulun mallintamisesta ilman parametreja:

1. Piirretään monikulmio, jonka koko vastaa haluttua kiulun pohjaa ja sivujen määrä lautojen lukumäärää.
2. Pursotetaan ensimmäisenä luotu monikulmio kiulun pohjaksi.
3. Tehdään monikulmiosta kopio, joka on alkuperäistä suurempi, niin että päällekkäin asetettuna niiden väliin jäävä tila määrää lautojen vahvuuden.
4. Erotetaan kuviosta laudat piirtämällä viivat toisiaan vastaavien kulmien välille.
5. Pursotetaan kaikki laudat haluttuun korkeuteen.
6. Pursotetaan yksi lauta kahvan pituuden verran muita korkeammaksi.
7. Piirretään profiili, jolla leikataan kahvan pää muotoon.



Edellisen mallin muokkaaminen on mahdollista; mm. kokoa ja eri osia voidaan muuttaa. Sen sijaan lautojen määrän tai pohjan muodon muuttaminen vaatii aloittamaan alusta. Edellisestä poiketen, parametrisesti mallinnettaessa työvaiheet dokumentoidaan komentosarjoiksi. Tämän ansiosta jokaista vaihetta voidaan palata muokkaamaan. Mallinnusohjelmalle siis kerrotaan kokonaisuutena mitä tehdään ja työvaiheita muokattaessa ohjelma luo mallin hetkessä yhä uudelleen. Parametrinen muotoilu on siis eräänlaista ohjelmointia. Työvaiheet peräkkäin kuvattuna muodostavat työkalun tietyn asian tekemiseen. Tällaista työkalua kutsutaan etenkin tietotekniikan yhteydessä algoritmiksi. Kiulun mallintavassa algoritmissa parametreja voivat olla vaikkapa lautojen lukumäärä, kiulun korkeus ja profiliviiva, jolla kädänsija muotoillaan. Muuttujat eivät siis aina ole lukuja: tapauksesta riippuen muita muuttujia voivat olla vaikkapa kuvat, tekstit sekä totuusarvot, eli kyllä tai ei -vastaukset.

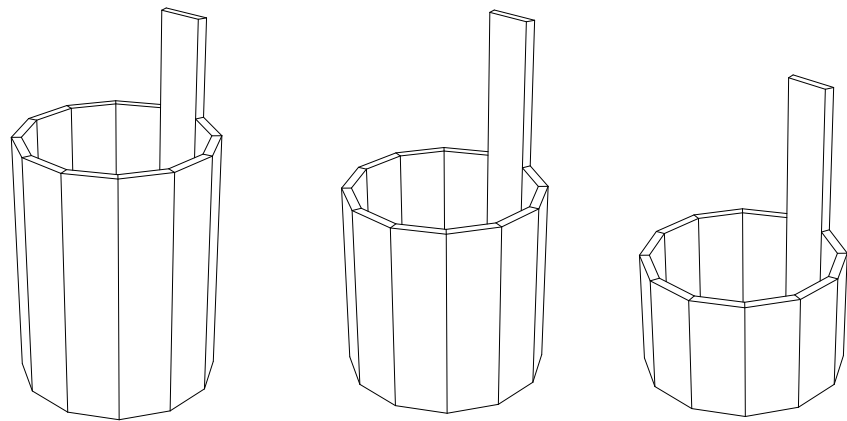


Parametrinen malli tuottaa nopeasti eri vaihtoehtoja muodosta ja nopeuttaa näin parhaan ratkaisun hakua ja kokeilua.

Parametrinen mallien ominaisuudet ja soveltuvuus

Parametriset mallit ovat kuin koneita, jotka tuottavat geometriaa ja muotoja. Niin algoritmien kuin koneidenkin hyöty perustuu niiden tarkoitukseen ja kykyyn toistaa asioita. Ompelukone tikkaa hetkessä tikin toisensa jälkeen ja sekunneissa syntyy sauma, jonka käsin ompeluun menisi tunteja. Samaan tapaan algoritmi toistaa vaikkapa seinäpinnan säännöllistä kuviota säästämällä mallintajalta jokaisen osan erikseen asettelun vaivan.

Monet koneet ovat käyttäjän säädettävissä. Ompelukoneen tikin mittoja ja tyyliä voi muuttaa ja eri asetuksin ommeltaessa jäljet voidaan aina tunnistaa ompelukoneen saumoiksi. Edellä esitetty parametrinen malli tuottaa tunnistettavia kiuluja, jotka voivat olla toisistaan poikkeavia. Näin todettakoon myös algoritmien avulla voitavan luoda joustavasti yhtenäisiä muotojen sarjoja.



Parametrinen mallin avulla muotoiltu tuotepöytä

Muunneltavuus on parametrinen mallien suuri vahvuus, koska mallin voi säätää tarpeeseen sopivaksi. Mallin asetuksien vaihtelu on nopea keino hakea parhaimpia mittasuhteita ja ominaisuuksia. Silti, kuten kaikkea, myös säätömahdollisuuksia voi olla liikaa, sillä lukuisten parametrien kohdalleen säätäminen voi työläydellään syödä mallin hyödyn. Lisäksi säätömahdollisuuksien kirjolla on helposti muodon suhteen sokeuttava vaikutus. Tämän välttämiseksi mallin käyttötarkoitus kannattaa rajata, jotta käyttö olisi selkeää ja tehokasta. Jos malli on liian monimutkainen, parametreja voidaan vähentää muuttamalla niitä vakioiksi. Muuttujia voidaan myös sitoa toisistaan riippuvaisiksi. Haluttuja tuloksia silmällä pitäen parametreille kannattaa antaa harkitut vaihteluvälit, jotka rajoittavat mahdollisten arvojen joukkoa ja estävät vääriä tuloksia. Mallin ominaisuuksia voidaan myös piilottaa tilapäisesti ja näin jakaa muodon määrittelyä eri vaiheisiin. Laajan säädettävyyden sijaan on usein järkevämpää tehdä algoritmista eri tarkoituksiin kohdennettuja muunnoksia.

Algoritmeja on mahdollista luoda monen tasoisin tehtäviin. Yksinkertaisimmillaan ne voivat olla pieniä työkaluja tavallisen mallinnuksen avuksi. Näin ne ovat verrannollisia vaikkapa saksisiin tai ruuvimeisseliin, jotka ovat käyttötarkoituksissaan verrattomat. Hiukan laajemmat algoritmit voivat tuottaa esimerkiksi osia, joita tarvitaan usein jonkin tuotteen mallinnuksessa. Perusteellisimmat algoritmit synnyttävät valmiita kolmiulotteisia malleja ja saattavat lisäksi tuottaa mittoja, ohjeita ja piirroksia tuotteen valmistusta varten. Tällainen malli ei välttämättä vaadi muokkaajaltaan juuri mallinnustaitoja, koska muokkaus voi olla vaikkapa vain lukujen antamista parametreiksi.

Algoritmin laajuus tai monimutkaisuus voi olla piilevää. Toisin sanoen yksinkertaiselta vaikuttava lopputulos saattaa vaatia syntyäkseen verrattain monimutkaisen prosessin. Monet itsestään selviltä tuntuvat asiat täytyy määrittellä erittäin tarkasti, jotta ohjelma käsittelee niitä halutulla tavalla. Mainitun monimutkaisuuden takia kannattaa harkita, mihin tehtäviin algoritmeja on järkevää luoda. Jos ne ovat verrannolliset koneisiin, myös niiden kehitystyötä voi verrata koneen rakennukseen. Molemmat ovat usein melko pitkäjänteisiä prosesseja, jotka vaativat perehtyneisyyttä työkaluihin ja tarpeisiin. Niin koneita kuin algoritmijakin on kannattavinta rakentaa toistuvien tehtävien. Mallinnustehtävä on hyvä jakaa osiin ja arvioida erikseen niiden sopivuutta algoritmiin. Perinteistä tapaa mallintaa voi käyttää muodon luonnosteluun ja työvaiheiden arviointiin ennen algoritmin luomista. On hyvä ajatella mallinnusta kokonaisuutena ja pohtia muuttuuko työvaiheiden luonne parametrien muuttuessa. Algoritmien ja käsin mallinnuksen yhdistely osoittautuu monesti hyväksi ratkaisuksi, sillä niiden työkaluvalikoimat ovat keskenään erilaiset ja siksi täydentävät toisiaan. Parametrinen malli voi tuottaa esimerkiksi muodon aihion, joka viimeistellään käsin. Mallinnusohjelmilla on myös rajoitteensa, joita voi ohittaa algoritmeilla. Parametriset mallinnustyökalut ovat tätä kirjoitettaessa vielä ohjelmien erikoisominaisuuksia. Useilla ohjelmilla on juuri niille ominaisia työkaluja ja toimintaperiaatteita. Näistä syistä ohjelman valinnalla on merkitystä työn sujuvuudelle ja työvaiheissa voi olla tarkoituksen mukaista käyttää eri ohjelmia.

Parametriset mallit ovat kuin reseptejä, jotka säilyttävät ruokia ja leivonnaisia valmistusohjeiden muodossa. Algoritmit ovat kolmiulotteisten mallien valmistusohjeita ja ne kertovat kuinka asioita toistetaan, mutta ne eivät sisällä toistojen tuloksia. Tämän takia ne reseptien tapaan säilövät tietoa perin tiiviisti. Mallinnusohjelmat ja niiden työkalut kehittyvät alati. Siksi parametrinen mallien säilyttäminen saattaa vaatia niiden luomiseen käytettyjen ohjelmaversioiden tai niissä käytettyjen työkalujen säilyttämistä. Kolmiulotteisten mallien tiedostomuodot ovat vakiintuneempia ja vanhojen mallien voidaan olettaa melko varmasti toimivan uusissa ohjelmissa.

Historiaa

Sana parametrinen on peräisin matematiikasta, mutta on kyseenalaista milloin muotoilijat ja arkkitehdit aloittivat sen käytön (Davis 2013). Matematiikassa parametriseksi kutsutaan suureita, jotka ovat tarkasti määriteltävissä funktioiden avulla. Funktioiden muuttujia voidaan kutsua parametreiksi (Weisstein 2003, 2150; Davisin 2013 mukaan). Parametrisessa muotoilussa nuo suureet kuvaavat muotoja. Toisin sanoen parametrinen mallien muodot ovat funktioiden määrittelyä.

Ennen kuin tietokoneet olivat yleisesti saatavilla, parametrinen mallit olivat harvinaisia erikoistapauksia. Ne rakennettiin todellisista tarvikkeista ja muotojen funktiot olivat fysiikan lakeja. Arkkitehti Antoni Gaudi käytti matemaattisten laskelmien ohella molemmista päästään ripustettuja naruja ja painoja suunnitellessaan Colonia Guellin kappelia. Hän saattoi luoda eri versioita holvien rakenteista säätämällä yksittäisiä parametreja. Koska riippuvat narut asettuivat kaariksi luonnon lakien määrääminä, parametreja muutettaessa malli otti muotonsa automaattisesti säästämällä arkkitehdiltä laskennan ja staattisen mallin rakentamisen vaivan. (Davis 2013)



Gaudin ylösalaisin riippuva kappelin parametrinen malli. (<http://www.gaudicoloniaguell.org/en/gaudi-lab/his-method>)

Toisen maailmansodan jälkeen tietotekniikan kehitys innosti tieteentekijöitä. Erityisesti 1960- ja 1970- luvuilla oli optimistisia ennustuksia tietokoneavusteisesta suunnittelusta ja muodon annosta. Tuon ajan tietokoneet olivat kuitenkin hintansa takia vain harvojen saavutettavissa ja muotoiluun niitä käyttivät lähinnä suurimmat auto- ja lentokonevalmistajat. 1980-luvulla henkilökohtaisten tietokoneet alkoivat yleistyä ja AutoCAD, yksi alan käytetyimmistä ohjelmista, julkaistiin. Sillä saattoi piirtää viivoja. Ensimmäinen kaupallisesti menestynyt parametrinen mallinnusohjelma Pro/ENGINEER julkaistiin 1988. Ohjelman kehittäjän Samuel Geisbergin mukaan parametrisen mallinnuksen tarkoitus on antaa suunnittelijalle mahdollisuus kokeilla eri vaihtoehtoja ja muuttaa mallin ominaisuuksia takautuvasti. (Davis 2013)

Vaikka AutoCADia ja monia muita mallinnusohjelmia ei lähtökohtaisesti tarkoitettu parametriseen mallinnukseen, niissä oli ominaisuus, joka mahdollisti algoritmien käytön (Davis 2013). Ohjelmien kehittäjät antoivat käyttäjille mahdollisuuden ohjata ja automatisoida ohjelmien toimintoja koodin avulla (Davis 2013). Tällaista käskytyä ainakin Toni Österlund ja Eero Lunden (2008:11) kutsuvat scriptaukseksi ja tähän tarkoitukseen luotuja koodikokonaisuuksia scripteiksi. Sanat ovat peräisin suoraan samaa asiaa tarkoittavasta englannin sanasta script. Scriptit, jotka ovat myös algoritmeja, ottavat vastaan parametreja, joita prosessoimalla ne luovat muotoja. Davisin (2013) mukaan tekstipohjaisen scriptauksen käyttöliittymät eivät ole kehittyneet merkittävästi sitten AutoCAD-ohjelman syntyänsä. Silti scriptauksesta on tullut yhä kiinnostavampaa, helpompaa ja tehokkaampaa, sillä tietokoneiden kasvava laskentateho mahdollistaa monimutkaisempia algoritmeja. Mallinnusohjelmien ominaisuudet ovat myös kehittyneet tarjoten scriptaajille uusia työkaluja. Lisäksi ohjelmointikielten kehitys käyttäjäystävällisempään suuntaan ja mallinnusohjelmien tuki uusille kielille laskevat kynnystä luoda algoritmeja.

Scriptauksen lisäksi on kehitetty käyttöliittymiä, joita voi pitää myös ohjelmointikielinä. Kyse on visuaalisista käyttöliittymistä, joissa ohjelman komentosarjat esitetään diagrammeina tekstin sijaan. Opinnäytetyöni muotoiluosuudessa käytän Grasshopper-ohjelmaa, joka toimii Rhinoceros-mallinnusohjelman rinnalla ja käyttää sen

ydintoimintoja ja työkaluja. Se on verrattain uusi ohjelma: Davisin (2013) mukaan se julkaistiin vuonna 2007 nimellä Explicit History. Se on myös saanut huomiota muotoilijoiden ja arkkitehtien keskuudessa.

Saarikorpi Design Oy:n toimeksianto

Suoritin korumuotoilijan tutkintoon kuuluvan työharjoittelujakson Saarikorpi Design Oy:ssä. Ensimmäinen tehtäväni oli mallintaa erään sormusmallin kaikki myytävät koot yksi kerrallaan Rhinoceros 5 -mallinnusohjelmalla. Työ kävi äkkiä tylsäksi toistoksi ja monessa vaiheessa huomasin kaipaavani tehokkaampia työkaluja. Oletin jonkun jo minua ennen olleen vastaavassa tilanteessa ja siksi myös parempia työkaluja kehitetyn. Hain tietoa ja opin. Käytin pian mallinnusohjelman makroja, eli toistettavia komentosarjoja, ja lisäosaa nimeltään Grasshopper. Tämän lisäosan avulla loin muutamasta yrityksen sormussarjasta parametriset mallit, joiden avulla saattoin mallintaa hetkessä halutun kokoisen sormuksen. Nämä mallit tuottivat myös tietoja, joiden avulla saattoi arvioida valmistuksen materiaalityyppejä ja kustannuksia. Yrittäjä Kristian Saarikorpi huomasi innostukseni ja ohjelman hyödyn ja ehdotti Grasshopperin käyttöä opinnäytetyössäni. Hän ehdotti tarkemminkin vihkisormusten suunnittelua, mihin myös suostuin.

Valinta ei silti ollut itsestään selvä, sillä Grasshopper aiheutti minussa uutuuden viehätystä ja houkutteli kokeilemaan rajojaan. Grasshopperilla on oma verkkoyhteisönsä, jossa ohjelman käyttäjät ja kehittäjät kohtaavat. Tuolla sivustolla ja muuallakin Internetissä olin nähnyt melko kokeellisia ja lennokkaita arkkitehtuuri- ja muotoiluprojekteja, joiden rinnalla ajatus vihkisormuksista tuntui rajoittuneelta. Harhauduin kuvittelemaan joutuvani silkkään käännöstyöhön, jossa rakentaisin algoritmin tuottamaan perinteisiä vihkisormuksia. Pohdittuani asiaa tajusin kuitenkin mahdollisuuteni tuottaa uutta muotoilua myös vihkisormusten joukkoon. Ohjelman hienous perustuu toistoon, mutta sen ei tarvitse tarkoittaa vain tuottavuuden lisäämistä ajankäyttöä tehostamalla. Uuden muotoilun luomiseksi toistoille ja mallin muuntautumiskyvylle on annettava uusia merkityksiä. Niin olivat tehneet muutkin kokeilijat, joita ehkä itseltäni salaa kadehdin. Tajusin myös olevani hyvässä asemassa, sillä verrattuna arkkitehtuuriin ja muuhun muotoiluun erikoistenkin korujen toteuttaminen on helppoa ja nopeaa. Lisäksi toimeksiantajallani oli jo valmiina kaikki laitteet, joita saattoin kuvitella tarvittavan muotojen toteuttamiseen. Olisin voinut suunnitella muunkin tyyppisiä koruja, mutta vihkisormukset ovat Saarikorpi Design Oy:n päätuotteita. Päätymällä muotoilemaan juuri niitä koen projektin olevan pelkkää kokeilua kokonaisempi, sillä erikoisetkin vihkisormukset ovat varteenotettavia tuotteita. Niitä hankitaan erityiseen tarkoitukseen ja ne ovat henkilökohtaisia esineitä, joiden toivotaan usein olevan yksilöllisiä. Koin saavani suunnitella tuotteita, joilla on lähtökohtaisesti suuri todennäköisyys päätyä tuotantoon ja käyttöön.

Työn tavoitteiksi muodostuivat Grasshopper-lisäosan käytön opettelu ja sen mahdollisuuksien tutkiminen sekä korun yksilöinti muodon muuntelulla. Asetin henkilökohtaiseksi tavoitteekseni perehtyä algoritmeihin ja yleensäkin parametriseen muotoiluun. Suurimman painon sai kuitenkin muotoilutyö ja Grasshopperin hyödyntäminen muodon annossa. Saarikorpi Design Oy:ssä käytetään mm. tietokoneohjattuja jyrsimiä ja 3D-tulostimia. Harjoitteluajanani opin käyttämään osaa laitteista, joten tietopohjani sormusten valmistusmahdollisuuksista oli jo valmiiksi melko hyvä. Suunnittelemani tuotteiden oli myös sovittava yrityksen valikoimaan.

Idean kehitys

Ennen päätöstäni suunnitella vihkisormuksia kartutin Grasshopper-taitojani kehittämällä kokeellisen mallin sormuksesta, joka perustui ideaan metallilangan pingottamisesta sormusrungon ympärille. Rungossa on tapauksesta riippuen vaihteleva määrä levyjä sormuksen sivuilta keskikohtaa kohti kohoavia ulokkeita. Ulukkeissa on kolot, joihin sormuksen ympäri kierretty lanka asettuu. Olin jo ennen opinnäytetyöni aiheen

valintaa tutustunut Eero Lundenin ja Toni Österlundin kurssityöhön ja selostukseen Algoritminen arkkitehtuuri. Sen ansiosta tiesin mahdollisuudesta käyttää satunnaisuutta muodon luomisessa. Olisin voinut soveltaa satunnaisuutta ulokkeiden sijaintiin, jotta ne asettuisivat sormuksen kehälle sattumanvaraisesti. Näin sormukset olisivat uniikkeja. Mallinsin sormuksen symmetrisesti asettuvien ulokkeiden, sillä aikaa kului lankojen mallinnuslogiikan rakennukseen. Lankojen mallinnus oli tarpeellinen vain idean esittämiseen, sillä tuotteen valmistusta varten malliksi riitti pelkkä sormusrunko, jonka päälle lanka kierrettäisiin.

Esitys mallista

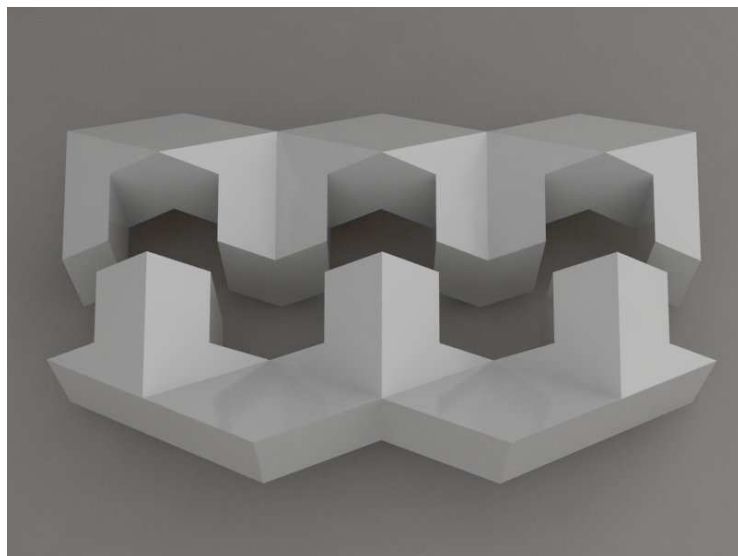
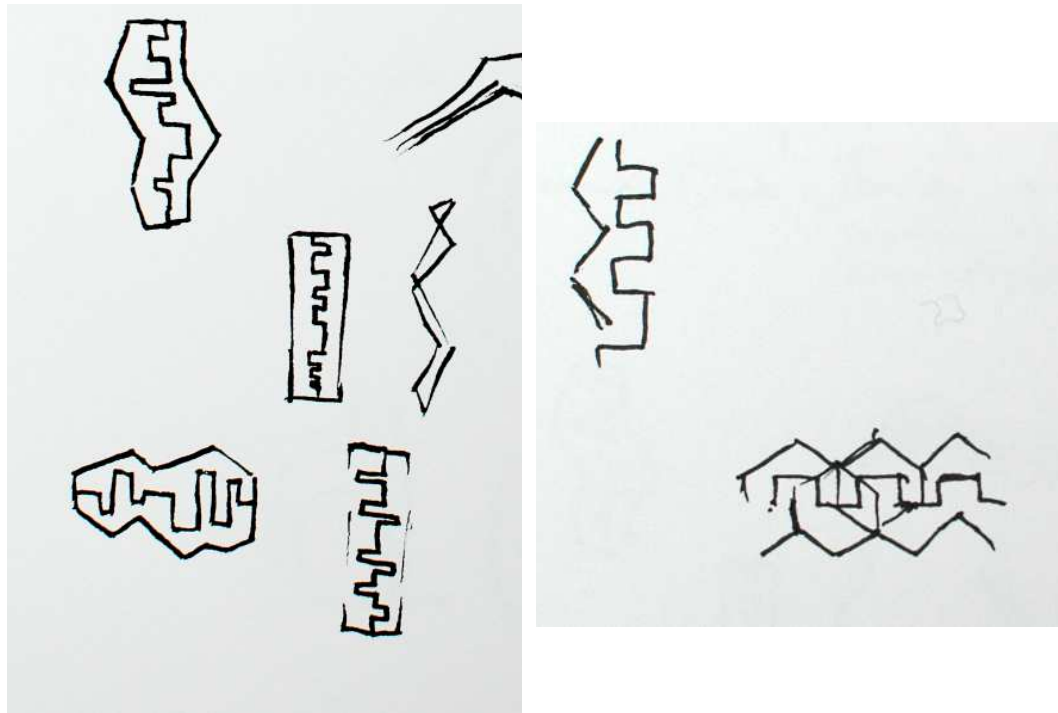


Esimerkki sormusrungosta ja ulokkeista, joissa on kolot langalle.

Esiteltyäni mallejani Kristian totesi niiden olevan turhan rajuja vihkisormuksiksi. Lisäksi vihkisormuksilta odotetaan muihin koruihin verraten melkoista mekaanista kestävyyttä. Langat olisivat saattaneet venyä tai kuluu poikki oletettavassa vuosikymmenten jatkuvassa käytössä. Lankojen päiden kiinnitys oli myös tekninen haaste, joka olisi vaatinut kokeiluja. Näistä syistä en kokenut idean jatkokehittelyä kannattavaksi opinnäytetyöni puitteissa.

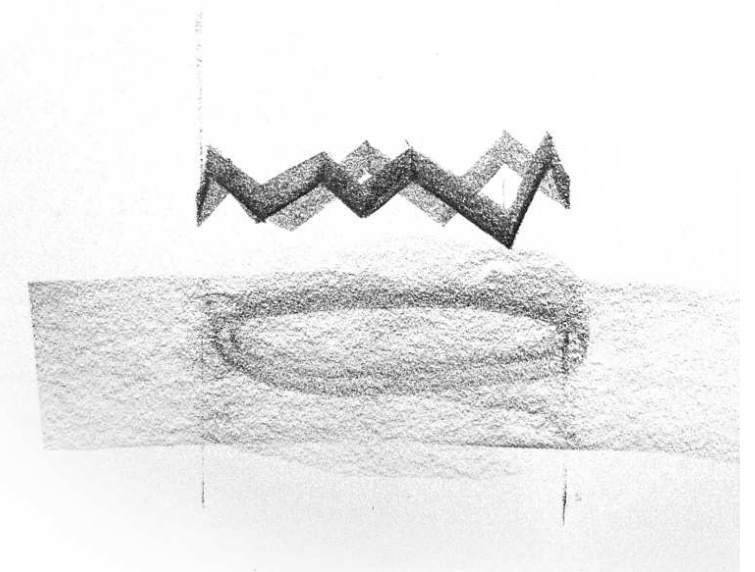
Seuraavaa ideaa kehitellessäni olin jo ajatellut asiani ja päättänyt suunnitella ensisijaisesti vihkisormuksia. Sain idean muotoilla sarjan vihki- ja kihlasormuksia, jotka ovat toisiinsa liittyviä pareja. Lisäsin vaatimukseen satunnaistekijän käytön, jolloin sormukset olisivat samankaltaisuudestaan huolimatta ainutlaatuisia. Ensimmäisiä ajatuksiani oli toteuttaa yksilöllisyys sormusten välisenä liittymäpintana, jolloin sormukset menisivät hieman sisäkkäin. Tämän idean pohjalta luonnostelemani muodot eivät olleet tyydyttäviä, sillä liittymäpinta ja sormuksen perusmuoto vaikuttivat toisistaan irrallisilta.

Luonnokset 1,2 ja 3: Liittymäpintoja



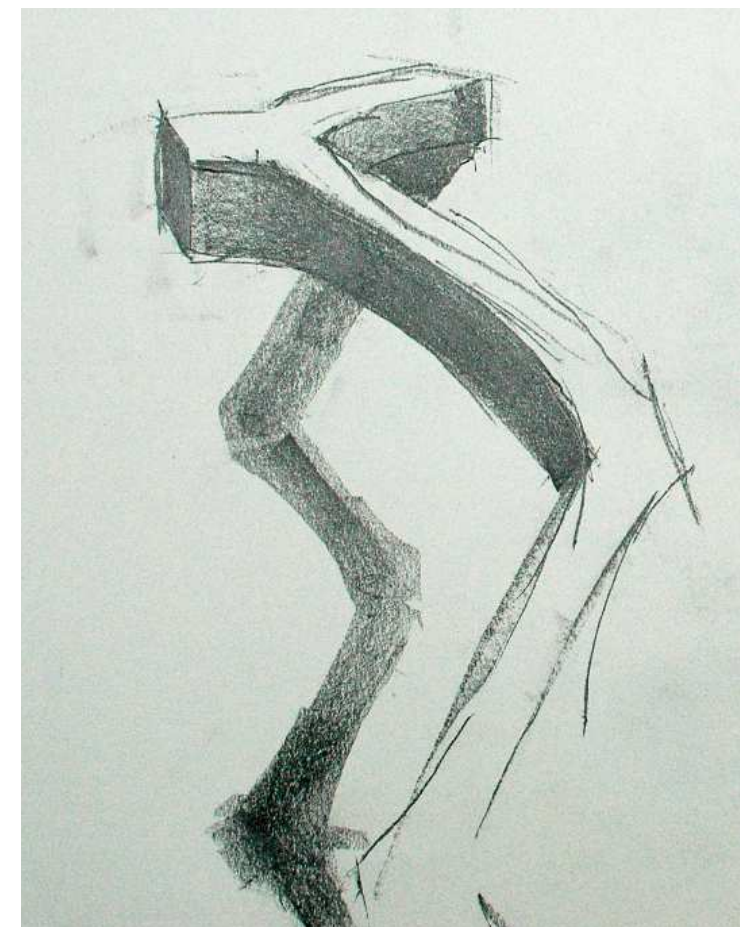
Siirryin pohtimaan sormuksen rakennetta ja sen saattamista jossain määrin satunnaiseksi. Luonnoksissani näkyvä sahalaitaviiva tuntui luontevalta lähtökohdalta sormuksen muodolle. Miellyttävin luonnoksistani muistuttaa hiukan kruunua tai oksaa. Tuo pieni luonnon kaltaisuus liitettynä satunnaisuuteen sai koko idean tuntumaan elolliselta.

Luonnos 4



Luonnos 5

Tämä luonnos miellytti minua eniten mm. jänteveytensä takia.

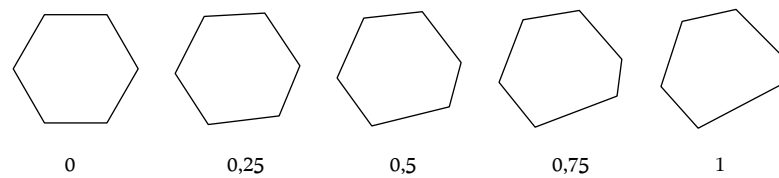


Malli

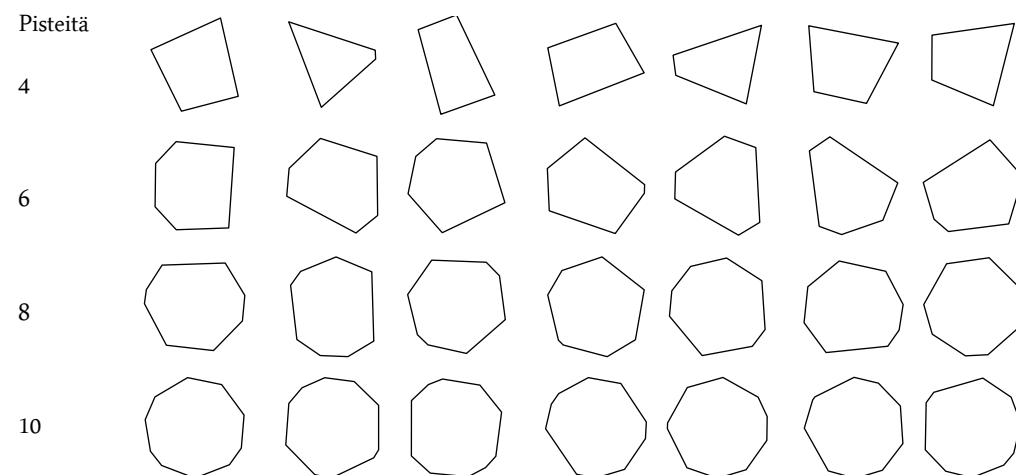
Tämä osio kertoo parametrisen mallin kehityksen vaiheista, mutta se ei ole rakennusohje. Sen tarkoitus on olla selostus muodonannosta luonnosteni pohjalta. Luomani malli on huomattava osa työni ideaa ja tarkoitusta. Sen tehtävä on toistaa tapauskohtaisesti määrittelemiäni muodon rakentumisen periaatteita lukemattomia kertoja. Niin kuin kaikessa muotoilussa, tekniikka asetti rajoja ja jouduin määrittelemään muodon osittain mallinnusohjelman ehdoilla. Siksi on luontevaa esittää muodon periaatteet mallin kautta.

Grasshopper ei ole ainoa keino tuottaa muotojen algoritmeja. Muitakin vastaavanlaisia ohjelmia on olemassa. Tutkin ja opiskelin työharjoitteluni aikoihin myös scriptausta eli eri ohjelmointikielten käyttöä mallinnusohjelman työkalujen hallintaan. Valitsin kuitenkin Grasshopperin visuaalisuuden ja siitä juontuvan intuitiivisuuden takia. Visuaalisuuden ansiosta ohjelmointia osaamatonkin saattaa hahmottaa eri komentojen kytkökset ja kokonaiskuvan mallin toiminnasta. Valintaani vaikutti myös melko varhainen tutustumiseni Grasshopperiin etsiessäni välineitä työharjoittelun sujuvoittamiseksi. Lisäksi käytin harjoittelussa Rhinoceros-ohjelmaa, jonka lisäosa Grasshopper on, ja jonka käytössä minulla oli jo valmiiksi taitoja.

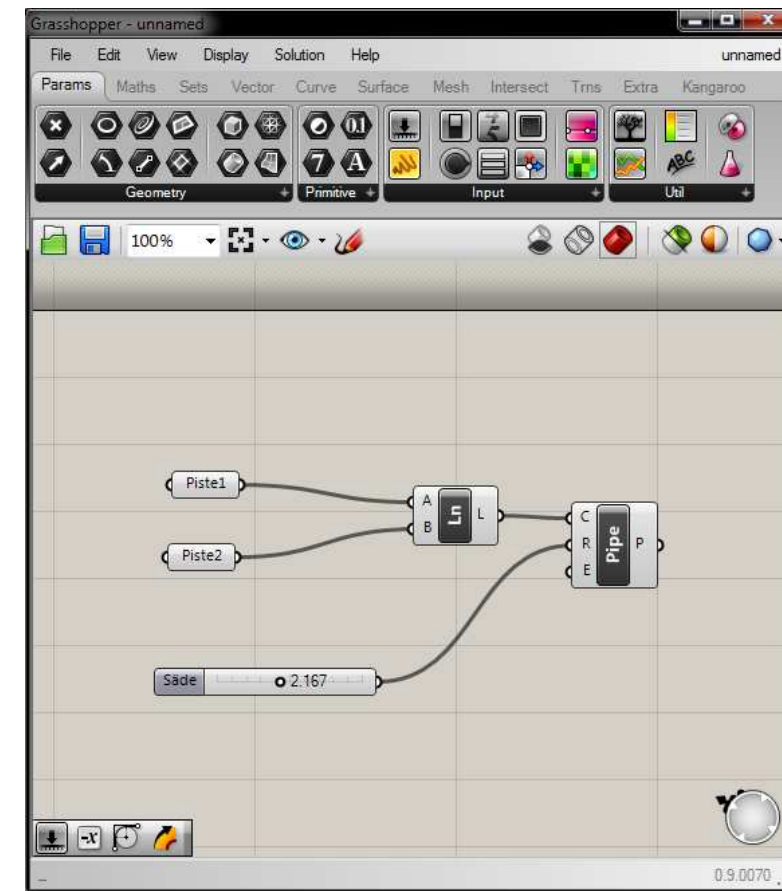
Aloitin kokeilut selvittääkseni pystynkö luomaan Grasshopper-ohjelmalla muotoja, jotka vastaavat tarpeeksi ideaani. Luonnoksen oksamaiset pykälät saavat paikkansa sahalaitakuvion kärkipisteissä. Tämän sidoksen takia koko sormuksen muotoa voidaan tehokkaasti säädellä niiden pisteiden kautta. Niihin minun oli luontevaa soveltaa satunnaistekijää luodakseni yksilöllisiä sormuksia. Käytin sahalaitaviivan lähtökohtana ympyrää. Sen kehällä on tasavälein pisteitä, joiden lukumäärä on yksi parametreista. Algoritmi arpoo jokaiselle pisteelle oman satunnaislukunsa, joka määrittää minkä verran pistettä siirretään ympyrän kehällä. Arvotut pisteet yhdistetään viivalla. Ennen kääntöä satunnaisluvut kerrotaan käyttäjän säätämällä pisteille yhteisellä tekijällä, joka on välillä [0,1]. Tämä luku määrää satunnaistekijän vaikutuksen voiman. Jos arvottu mittasuhteisto vaikuttaa liian rajulta tai symmetriseltä, kertoimella voi korjata tulosta haluttuun suuntaan. Arvolla 0 sormus on täysin symmetrinen ja ylöspäin säädettyä poikkeamat kasvavat.



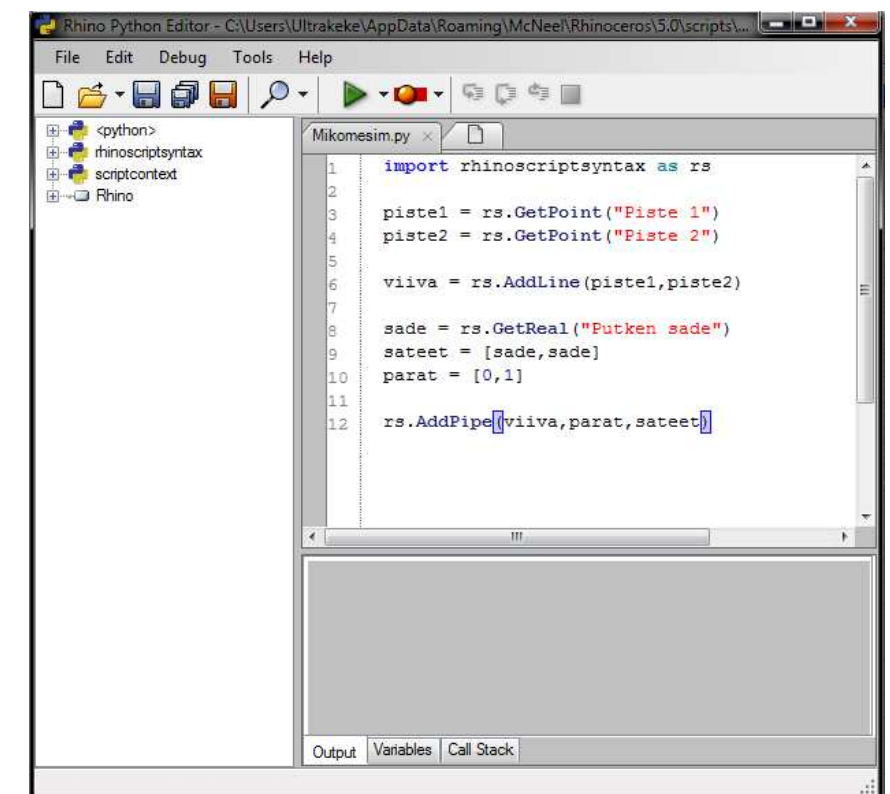
Esimerkki satunnaistekijän vaikutuksesta eri kertoimin



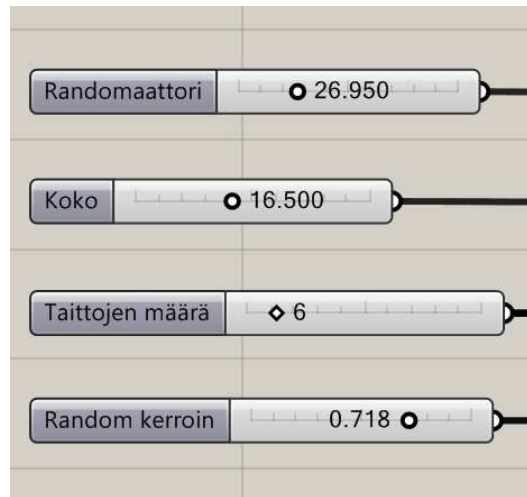
Yhdistetyt pisteet muodostavat monikulmioita, jotka ovat sormusten muodon lähtökohtia.



Grasshopperissa parametrien ja komentojen väliset sidokset kuvataan johtoina. Tämä algoritmi tuottaa putkimaisen pinnan kahden pisteen välille. Poikkileikkauksen säde ja päätepisteet ovat parametreja

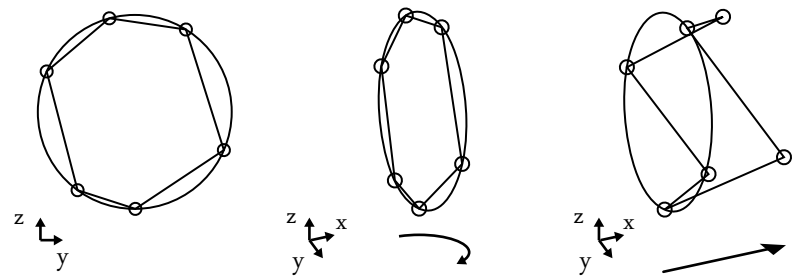


Vertailun vuoksi: Vastaavanlaisia putkia tuottava Python-koodikielellä luotu algoritmi Rhinocerosin koodityökalussa.

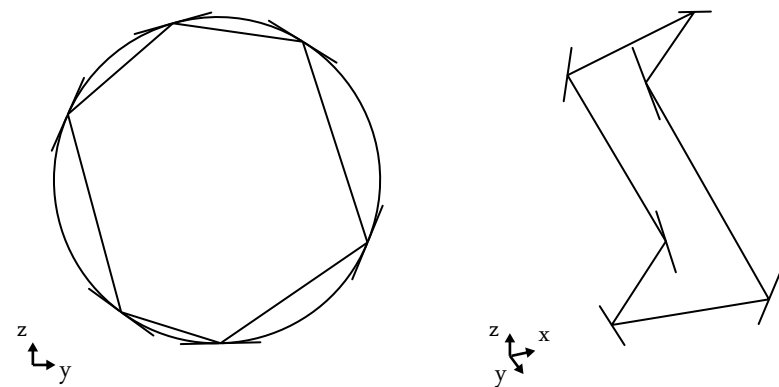


Liukusäätimet ovat Grasshopperin kätevimpiä komponentteja. Säädettäessä malli päivittyy reaaliaikaisesti, mikä kuuluu ohjelman merkittävimpiin ominaisuuksiin. Kuvan säätimet määräävät sormuksen koon, sahalaitaviivan kärkipisteiden määrän, satunnaisluvun ja sen vaikutuskertoimen.

Sahalaitaviiva muodostuu siirtämällä joka toinen piste sivuun ympyrän kehältä. Siirto määrää myös sormuksen leveyden. Tähän vaiheeseen kehitetty malli muistutti jos melkoisesti sormusta luonnoksessa 4. Pyrin kohti luonnos 5:n muotoja, mutta sahalaitaviiva mahdollisti muitakin muotokokeiluja. Loin erilaisin periaattein geometriaa tuon viivan ympärille.



Tavoittelin muotoa, jossa sahalaitaviivan kulmissa on pykälät, jotka ovat kuin oksan haara- tai silmukohtia. Pykälän leveyttä merkitsemään sijoitin viivan kärkipisteisiin lyhyet janat. Ne ovat muodon lähtökohtana käytetyn ympyrän vastaavien pisteiden tangenttien suuntaiset.

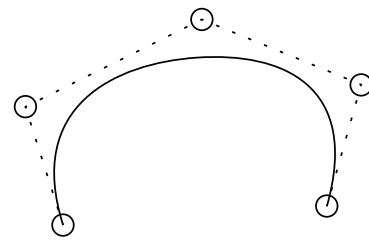


Tämä kokeilu syntyi lähes suoraan sahalaitaviivasta.



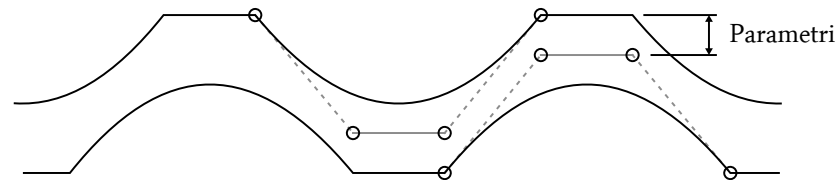
Tämä versio taas muistuttaa hiukan Saarikorpi Design Oy:n Wave-sormuksia.

Sormuksen profiilin muodostamiseksi janojen päätepisteiden välille täytyi luoda viivoja. Tähän käytin NURBS-käyriä, joita voidaan muodostaa ja hallita pisteiden avulla. Käyrille määritetään alku- ja loppupisteet sekä sarja hallintapisteitä, jotka määräävät muodon alun ja lopun välillä. Nuo pisteet vetävät niitä vastaavia käyrän osia puoleensa pingottaen ne jatkuvaksi kaarevaksi muodoksi.



Esimerkki: NURBS-käyrä pisteineen

Käytin pykälää merkitseviä janoja paitsi käyrien päätepisteinä, mutta myös välillisesti hallintapisteinä. Määritin janojen kopioiden päätepisteet käyrien hallintapisteiksi. Kopioitujen janojen etäisyys alkuperäisistä sijainneistaan on yksi mallin säädettävistä parametreista, joka vaikuttaa profiilin vahvuuteen ja siksi merkittävästi sormuksen olemukseen.

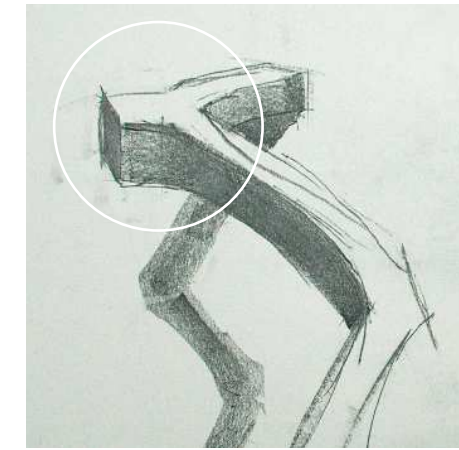


Osa sormuksen profilia levitettyä tasoon

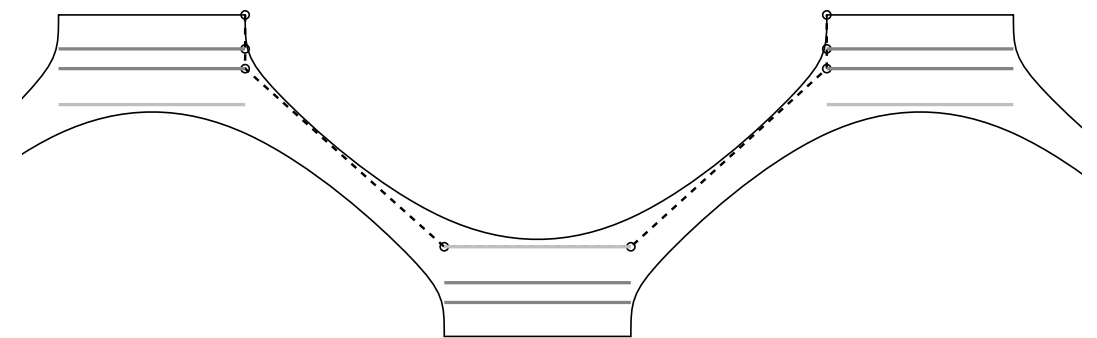


Sormus, jossa on sama profiilin periaate.

Edellinen muoto vaikutti vielä turhan kömpelöltä, joten lisäsin muodon määritteitä. Pykälän sijaintia ja leveyttä merkitsevät janat olivat edelleen oiva lähtökohta tavoitellessani tarkempaa pykälän muotoa. Viimeksi esitettyssä mallissa janojen kopiot taivuttavat pykälän väliset kaaret suurin piirtein keskeltä. Näiden lisäksi määritin malliin vielä kahdet kopiot, joiden päätepisteet hallitsevat käyrien päiden seutuja. Näiden uusien janojen sijainti on edellisten kopioiden tapaan käyttäjän säädettävissä.



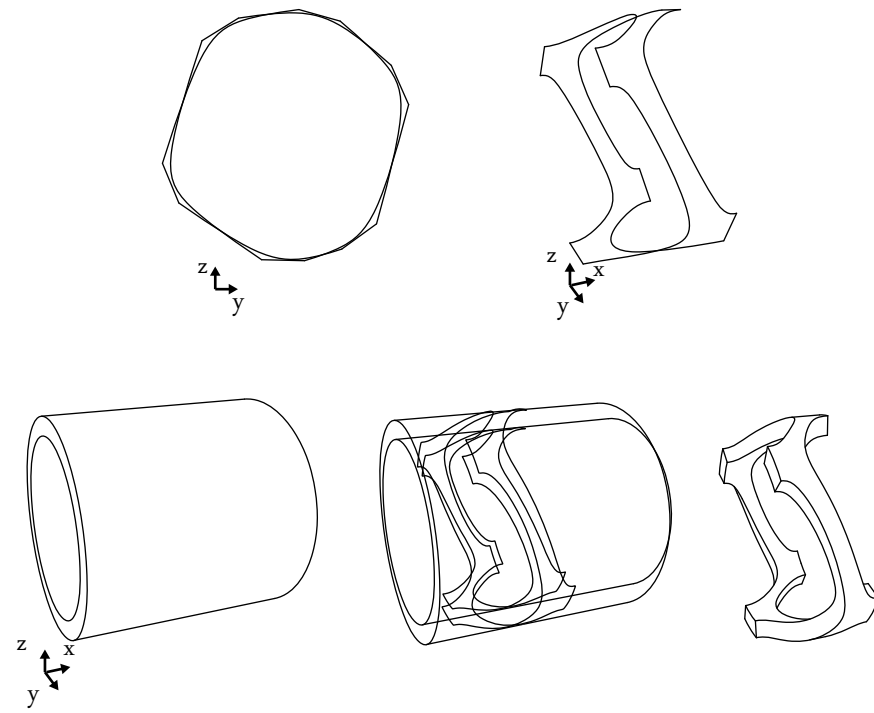
Tavoittelemani muoto



Käyrän muodostus uusilla hallintapisteillä

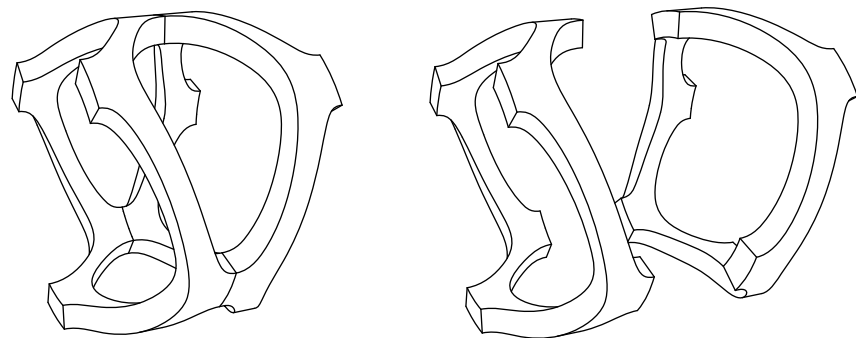
Näillä parametreilla profiilin muoto vastasi vaatimuksiani. Arvioin tätä tarkemman määrittelyn olevan turhaa sekä muodon että mallin käytettävyyden kannalta.

Profiilin määrittelyn jäljiltä sormuksen muoto sivusta katsoen muistuttaa vielä melkoisesti lähtökohtana käytettyä monikulmiota. Lisäksi profiilille on annettava vahvuus, jotta kyseessä olisi kolmiulotteinen kappale. Määritin malliin kaksi sisäkkäistä sylinteripintaa, joille profiili heijastetaan. Tästä syntyneet projektiot ikään kuin leikkaavat kahden sylinteripinnan muodostamasta putkesta tavoitellun sormuksen muotoisen kappaleen. Putken vahvuus siis määrää sormuksen paksuuden säteen suunnassa.



Profilin projektio ja sormuksen pintain muodostus

Kun tiesin haluamani muodon olevan mahdollinen, lisäsin algoritmiin sormukselle parin. Pykälät olivat ilmeisin kohta liittää sormukset toisiinsa. Tein sormuksesta kopion, joka peilautuu alkuperäiskappaleestaan toisen puolen pykälien määräämän tason suhteen. Jotta sormukset olisivat uniikkeja myös keskenään, käytin toista satunnaistekijää kääntämään peilatus sormuksen toisen puolen pykälää. Peilaustasoa vastaavat pykälät pysyvät käännössä paikallaan.



Muodon määrittely eteni intuitiivisesti lähtien suurpiirteisimmästä eli ympyrästä, edeten vaiheittain mittakaavaa tarkentaen ja päätyen pykälien viereisiin kaariin. Rhinoceros ja Grasshopper käyttävät muotojen luontiin samoja ydintoimintoja. Ohjelmien käyttäjille näyttäytyvät työkalut kuitenkin poikkeavat toisistaan, koska ne soveltavat ydintoimintoja hiukan eri tavoin. Grasshopper ei ole täysin valmis ohjelma ja jotkut työkalut eivät toimi kaikissa oletettavissa tilanteissa. Nämä seikat aiheuttivat suurimmat hankaluuteni algoritmia rakentaessani.

Algoritmin avulla luotuja malleja





Valmiiksi kelpuuttamassani mallissa ovat käyttäjän määrättävissä seuraavat parametrit:

- Sormuksen läpimitta eli koko
- Pykälien lukumäärä
- Ensimmäisen sormuksen leveys
- Toisen sormuksen leveys
- Satunnaistekijän voimakkuus
- Toisen satunnaistekijän voimakkuus
- Pykälien leveys
- Käyrää hallitsevan pisteparin etäisyys vastaavan pykälän kärjestä
- Kaksi parametria pykälän kylkimuodon hallintaan

Tuloksia ja kehitysideoita

Valmistin kokeeksi neljä hopeista sormusparia algoritmini tuottamista malleista. Jyrsin mallit koulun Roland JWX-30 -vahajyrsimellä ja valatin ne Kultataide Oy:ssä. Viimeistelin valetut sormukset itse. Kokeilin ja kokeilutin sormuksia tovereillani. Kulmikkaista muodoista huolimatta sormukset vaikuttivat käyttökelpoisilta, sillä terävät särmät pyöristyivät sopivasti kiillotuksessa.



Sormusten muotoja vahajyrsimen jäljiltä



Kävin läpi valettujen kappaleiden pinnat
käsityökaluin.

Viimeistelyjä sormuksia



Olen tyytyväinen mallin tuottamiin muotoihin. Sillä saa aikaan yhteneviä, mutta melko erilaisia sormuksia sekä miehille että naisille. Tämän toteamiseksi koesarjassani on siroja ja hieman rotevampia sormuksia.



Esimerkit siroista ja jykevämmistä muodoista



Mallissani on melko paljon säädettäviä parametreja. Sormusmallin tuotteistamiseksi ja valintoja helpottamaan säätimiä voisi ryhmitellä. Mallista voisi tehdä myös eri versioita, joissa eri säätimiä olisi lukittu tapauksesta riippuen. Näin valintoja olisi tehty jo valmiiksi. Mietimme yrittäjä Krisitan Saarikorven kanssa mallin tuotteistusta ja mahdollisia asiakaspalvelutilanteita. Sormuksia voitaisiin esimerkiksi mallintaa asiakkaan läsnäollessa, mikä korostaisi sekä tuotteen että palvelun yksilöllisyyttä. Sormussarjan idea satunnaisuudesta on arvo, jonka havainnollistaminen asiakkaille on tärkeää.

Kristianin mielestä sormukseni sopivat yrityksen tarjontaan erityisen hyvin. Hänen mukaansa asiakkaat valitsevat Saarikorpi Designin sormuksia perinteisestä poikkeavan muotoilun ansiosta. Yrityksen jatkuvasta toiminnasta päätellen muodot eivät ole liian erikoisia tai vaikeasti lähestyttäviä. Saarikorpi Design erottuu korualan yrityksistä myös hyödyntämällä melko ennakkoluulottomasti uutta tekniikkaa. Täten vakuutuin onnistuneeni, sillä myös muotoiluni tekninen luonne ja sormusten valmistustapa sopivat yrityksen toimintaan.

Työni on myös todiste Grasshopperin hyödyllisyydestä ja käytettävyydestä. Algoritminen muodonanto on uusi toimintatapa korualalla ja tässä tapauksessa se mahdollisti uudenlaisen korukonseptin liitettäväksi Saarikorpi Design Oy:n kokoelmiin. Satunnaisuuden hyödyntäminen ei ole uusi tapa tuotteiden sarjallistamisessa, mutta parametrinen malli yhdistää satunnaisuuden verrattain uusiin tapoihin määrittää muotoja ja valmistaa koruja.

Sormussarjani nimeäminen osoittautui hankalaksi, sillä tässä tapauksessa en halua ohjata mahdollisten asiakkaiden mielikuvia liikaa, jottei ostajien joukko rajautuisi sopimattomien arvolatausten takia. Vaikka läpi projektin mielessäni ovat säilyneet mielikuvat oksasta ja kruunusta, niiden käyttäminen nimessä olisi ollut yksipuolista ja liian suoraviivaista. Sarjan nimi voi hyvinkin muuttua tuotteistamisen myötä, mutta toistaiseksi se olkoon Sattuma.

Lähteet

Davis, D. 2013. A History of Parametric. <http://www.danieldavis.com/a-history-of-parametric/>

Lunden, E. & Österlund, T. 2008. Algoritminen arkkitehtuuri. <http://www.osterlund-ark.fi/bookshelf/>