

168542

YILDIZ TEKNİK ÜNİVERSİTESİ
FENBİLİMLERİ ENSTİTÜSÜ

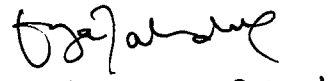
ERKEN TASARIM
SÜRECİNDE TÜRETİCİ SİSTEMLERİN
KULLANIMI


Saro DİONYAN


FBE Mimarlık Anabilim Dalı Bilgisayar Ortamında Mimarlık Programında
Hazırlanan

YÜKSEK LİSANS TEZİ

Tez Danışmanı: Yrd. Doç. Dr. Birgül ÇOLAKOĞLU


Prof. Dr. Oya Pakdi


Yrd. Doç. Dr. Birgül ÇOLAKOĞLU


Prof. Dr. Güler Çoğdas

İSTANBUL 2005

İÇİNDEKİLER

	Sayfa
ŞEKİL LİSTESİ	iv
ÇİZELGE LİSTESİ.....	vi
ÖNSÖZ.....	vii
ÖZET.....	viii
ABSTRACT	ix
1. GİRİŞ.....	1
1.1 Tezin Yapısı.....	2
2. ERKEN TASARIM SÜRECİ.....	3
2.1 Tasarım Sürecinde Tasarımcı Davranışları ve Araçları	4
2.1.1 Eskiz'in Tasarım Geliştirmedeki Kullanımı.....	5
2.1.2 Tasarım Sürecinde Analoji.....	6
2.1.3 Tasarım Sürecinde Şekil Belirmesi (Shape Emergence).....	7
3. TÜRETİCİ SİSTEMLER ve TASARIM	9
3.1 Türetici Sistem Örnekleri	10
3.1.1 Parametrik Tasarım Araçları	10
3.1.1.1 Bir parametrik mimari tasarım aracı: Revit.....	11
3.1.2 Biçim Gramerlerini Kullanan Türetici Sistemler	12
3.1.2.1 İki Boyutlu Biçim Türetme Aracı: Shaper 2D (McGill, 2001)	13
3.1.2.2 Üç Boyutlu Biçim Türetme Aracı: 3D Shaper (Wang ve Duarte 2001)	13
3.1.2.3 Biçim Gramerleriyle Stil Türetimi: Archi-Dna (Kwon 2003).....	14
3.1.3 Evrimsel Algoritmalara Dayalı Türetici Sistem Örnekleri.....	15
3.1.3.1 Bağımsız Bir Evrimsel Sistem: Argenia.....	16
3.1.3.2 Evrimsel Bir Eklenti (Plug-in) Örneği: Genr8	18
3.1.4 Kısıtlama Tabanlı (Constraint-Based) Tasarım Sistemleri.....	19
3.1.4.1 Sanal Gerçeklikte Türetim: Smart Objects.....	20
3.1.4.2 Bir 3ds Max Eklentisi: Xp-Gen.....	20
3.1.5 Mimari Uygulamada Türetici Sistem Kullanımına Bir Örnek	21
4. BİR TÜRETİCİ SİSTEM DENEMESİ: ConGen	24
4.1 ConGen Arayüzü	25
4.2 Alt-birimlerin İsimlendirilmesi	31
4.3 Alt-birimlerin Renklendirilmesi	32
4.4 Alt-birim Boyutlarının Ayarlanması	33
4.4.1 Hedeflenen Boyut Değerlerinin Türetimi İçin Programa Kod Eklenmesi	34
4.4.2 Max Programında Üretilmiş Şekillerin ConGen Nesneleriyle İlişkilendirilmesi..	35
4.5 Alt-Birim Pozisyonlarının Ayarlanması.....	36
4.6 Değiştiricilerin Uygulanması.....	37

4.7	ConGen Programının Kullanım Kısıtlamaları.....	40
4.8	ConGen Programı ile Tasarım Türetmek için Önerilen Çalışma Metodu.....	41
5.	CONGEN İLE ÜRETİLMİŞ TASARIM ÖRNEKLERİ	43
5.1	Ürün Tasarımı Örneği: Parfüm Kutusu	43
5.2	Ürün Tasarımı Örneği: Abajur	45
5.3	Mimari Şematik Tasarım Örneği: Bir Konut Planlama Çalışması.....	47
5.4	Önerilen Metodolojinin Kullanımı: Bir koltuk tasarımı.....	49
6.	SONUÇLAR.....	56
6.1	ConGen'in Kullanımı ve Geleceği	56
6.2	Türetici Sistemlerin Tasarımda Kullanımı ve Gelecekleri	57
	KAYNAKLAR.....	59
	İNTERNET KAYNAKLARI.....	61
	EKLER	62
	Ek 1 ConGen programının kaynak kodu.....	63
	ÖZGEÇMİŞ.....	94

ŞEKİL LİSTESİ

Şekil 3-1 Parametrik nesne kütüphanesi arayüzü (Revit web sitesi).....	11
Şekil 3-2 Proje dokümanlarının aynı arayüzde kullanımı (Revit web sitesi).....	12
Şekil 3-3 Shaper 2D programının arayüzü (Knight 1999)	13
Şekil 3-4 3D Shaper programının arayüzü (Knight 1999)	14
Şekil 3-5 Archi-Dna programının arayüzü. (Kwon 2003)	15
Şekil 3-6 Archi-Dna programıyla üretilmiş kentsel tasarım örneği (Kwon 2003).....	15
Şekil 3-7 Milano’da yapılacak yeni Fütürizm Müzesi için mevcut durum ve Argenia programıyla üretilmiş üç alternatif (Soddu 2004)	17
Şekil 3-8 New York kenti kimliğinden elde edilmiş kodla üretilmiş bir kent parçası (Soddu 2004)	17
Şekil 3-9 Genr8 tarafından üretilmiş dijital bir yüzey (kaynak: emergent design group web sitesi).....	18
Şekil 3-10 Bir CNC makinesi aracılığıyla üretilmiş, Genr8 yüzeyi (Hemberg ve O’Reilly 2004)	19
Şekil 3-11 Simetri kısıtlaması konulan cephede, ana aksın bir yanındaki duvarın hareket ettirilmesi, aksın diğer yanındaki duvarın pozisyonunu değiştirmektedir (Eggink, Gross, Do 2001).....	20
Şekil 3-12 Şömine kutusu, merkezdeki akstan uzaklaştıkça saydamlaşmaktadır (Eggink, Gross, Do 2001).....	20
Şekil 3-13 Xp-Gen üretimlerine bir örnek (Pak, Özener, Erdem 2003)	21
Şekil 3-14 Tasarım alternatiflerinden bir örnek maketi (Scheurer 2003)	22
Şekil 3-15 Geliştirilen programın arayüzü (Scheurer 2003)	23
Şekil 3-16 Tasarlanan platformun, alt kottan elde edilen, üç boyutlu perspektif çizimi (Scheurer 2003)	23
Şekil 4-1 ConGen programının akış diyagramı.....	25
Şekil 4-2 Katlanabilir panelin açık ve kapalı halleri	26
Şekil 4-3 Object01 adıyla etiketlenmiş bir parametre grubu.....	26
Şekil 4-4 Bir etiket örneği (pos. X).....	26
Şekil 4-5 Bir çevirici örneği.	26
Şekil 4-6 Bir düğme örneği.	27
Şekil 4-7 Renk seçim düğmesi ve renk seçimi arayüzü	27
Şekil 4-8 Değiştirilebilir metin kutucuğu örneği.....	27
Şekil 4-9 Onay kutucuğunun işaretlenmemiş ve işaretlenmiş durumu (limit effect).....	27
Şekil 4-10 İfo panelinin görünümü	28
Şekil 4-11 Boyutlandırma panelinin görünümü.	29
Şekil 4-12 Konumlandırma panelinin görünümü.....	30
Şekil 4-13 Değiştiriciler panelinin görünümü.	31
Şekil 4-14 Alt-birimler isimlendirilmeden önce ve sonra arayüz görüntüleri.....	32
Şekil 4-15 Alt-birimlerin renklendirilmesi ve arayüzün güncellenmesi	33
Şekil 4-16 Genişlik parametresi için minimum 30 maksimum 100 değeri girilen nesnenin, minimum, maksimum ve türetilebilecek değerler aralığının gösterimi.....	34
Şekil 4-17 ConGen’de serbest formların kullanılması	36
Şekil 4-18 Pozisyon sınırları için girilmiş değerlerin oluşturduğu sanal çerçeve.	36
Şekil 4-19 Bir küpe farklı oranlarda küreselleştirme değiştiricisinin uygulanmasına örnekler.....	37
Şekil 4-20 Bir prizmaya uygulanan bükme değiştiricisinin etkilerine örnekler.....	38
Şekil 4-21 Bir prizmaya uygulanan skew (çarpıtma) değiştiricisinin parametrelerinin etkilerine örnekler.	39
Şekil 4-22 Bir prizmaya uygulanan konikleştirme değiştiricisinin parametrelerinin etkilerine örnekler.	40

Şekil 5-1 ConGen'in türettiği alternatif tasarımlardan örnekler	44
Şekil 5-2 ConGen'in türettiği alternatif tasarımlardan örnekler	46
Şekil 5-3 ConGen'de türetilen, hepsi aynı mekân büyüklüklerine ve toplam alana sahip konut alternatifleri.....	49
Şekil 5-4 Farklı markalara ait koltuk tasarımlarından örnekler	50
Şekil 5-5 Örneklerin değerlendirilmesiyle elde edilen sınır değerler.....	51
Şekil 5-6 Boyut ve pozisyon değerlerine dayanarak türetilen tasarım alternatiflerinden örnekler	54
Şekil 5-7 Max değiştiricilerinin uygulanmasıyla türetilebilecek alternatiflerden örnekler.....	54
Şekil 5-8 Final tasarım sunumu örneği.....	55



ÇİZELGE LİSTESİ

Çizelge 5-1 Farklı min ve max değerleri için üretilmiş değer örnekleri	34
Çizelge 5-2 ConGen'e eklenilebilecek kod örnekleri ve bunların işlevlerine örnekler	35
Çizelge 6-1 Parfüm kutusu alt-birim parametrelerine atanan değerler	43
Çizelge 6-2 Uygulanan 3ds Max değiştiricileri için verilen sınır değerler	44
Çizelge 6-3 Abajur alt-birim parametrelerine atanan değer ve ek kodlar	45
Çizelge 6-4 Uygulanan 3ds Max değiştiricileri için verilen sınır değerler	46
Çizelge 6-5 Konut türetimi için kullanılan değer ve hedef kodlar tablosu	47
Çizelge 6-6 Koltuk türetimi için kullanılan değer ve hedef kodlar tablosu	51



ÖNSÖZ

Bu tez, Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsüne bağlı, Mimarlık Anabilim Dalı, Bilgisayar Ortamında Mimarlık Yüksek Lisans Programı için hazırlanmıştır. Konusu, erken tasarım sürecinde üretici sistemlerinin kullanımınıdır.

Bu çalışma sırasında gösterdiği anlayış ve destek için, tez danışmanım, Yrd. Doç. Dr. Birgül Çolakoğlu'ya teşekkür ederim.



ÖZET

Bu tez, erken tasarım sürecinde üretici sistemlerin kullanımını tartışmak amacıyla yazılmıştır. BDT (Bilgisayar Destekli Tasarım) yazılımları genellikle tasarım sürecinin son aşamalarında, detaylandırma, ya da sunum çalışmalarını hızlandırma amacıyla kullanılmaktadırlar. Ancak son yıllarda geliştirilen hesaplama dayalı tasarım (computational design) programları, bilgisayarların tasarım konsepti ya da form geliştirme sürecinde de kullanılabilceğini göstermektedirler. Üretici tasarım sistemlerinin kullanımı konusunda öne sürülen çeşitli fikirlerin ve örneklerin etkilerini tartışabilmek amacıyla geliştirilmiş olan CONGEN programı, farklı tasarım amaçları için kullanılabilcek, tasarımcının belirlediği sınırlamalar içerisinde form alternatifleri üretebilecek bir üretici sistem örneğidir. Bu program aracılığıyla yapılan çalışmalar ve farklı kullanıcılardan alınan tepkiler bu tür yazılımların geleceğin tasarım dünyasında önemli bir yere sahip olabileceğini göstermektedirler.

Anahtar Kelimeler: Üretici Tasarım Sistemleri, 3ds Max



ABSTRACT

This thesis discusses the use of generative design tools in the early design phases. Available CAD (Computer Aided Design) softwares are usually developed for the final steps of design process. But the new computational tools for design, proves that the computer technology can also be used in concept or form developing phases of design. The developed plug-in (ConGen) used to test the advantages and disadvantages of generative design tools in the early phases of design. ConGen is a generative design tool which generates parametric form alternatives between user-specified constraints. Generated designs with ConGen and the user feedbacks show that this type of tools will have an important role in the future of design industry.

Keywords: Generative Design Tools, 3D Studio MAX.



1. GİRİŞ

Bilgisayar destekli tasarım ve mühendislik programlarının ortaya çıktığı yetmişli yıllardan itibaren, tasarım teorisi ve metodolojisi alanında yapılan çalışmalar, tasarıma bakışımızı değiştirdi (Chan, 1998). Dijital tasarım araçları, günümüzün tasarım ve üretim süreçlerinin vazgeçilmez bir parçası haline gelmişlerdir. Ancak geliştirilen araçlar, tasarım sürecinin daha iyi anlaşılması, sistematize edilebilmiş aşamalarına (üretim, detaylandırma, maliyet analizi, vb.), başarılı bir şekilde entegre olmalarına rağmen, daha az anlaşılabilen erken tasarım aşamalarına gerçek bir katkıda bulunabilecek şekilde kullanılamamaktadırlar.

Son yıllarda, bu alandaki eksikliği gidermek amacıyla, hesaplama dayalı tasarım (computational design) konusunda birçok yeni araştırma yapılmaktadır. Türetici tasarım sistemleri deyimi, bu alanda yapılmış farklı çalışmaları kapsamaktadır. Biçim gramerleri, parametrik tasarım, evrimsel algoritmalar ve yeniden yazma kurallarına dayanan birçok yeni yazılım, türetici tasarım sistemlerin kullanımını yaygınlaştırmaktadırlar.

Bu tür sistemler, temel olarak, tasarımcının girdiği basit verilere dayanarak çözüm geliştirebilecek şekilde kurgulanırlar. Belirli bir tasarım problemi için çözüm geliştirmek isteyen tasarımcı, bu noktada devreye girecek türetici sistemlerin yardımıyla, kriterlerin çizdiği sınırlara uyan alternatifler oluşturabilir. Ayrıca türetici sistemler, farklı temsil yöntemlerini kullanarak tasarımcıda bir kavrayış değişikliği sağlayabilir ve yeni fikirlerin doğmasına yardımcı olabilirler.

Erken tasarım sürecinde türetici sistemlerin kullanımını inceleyebilmek için bir parametrik form türeticisi olan ConGen programı geliştirilmiştir.. Bu tez, ConGen'in tasarım sürecine ne gibi katkılar sağlayabileceğini tartışmaktadır. Elde edilen sonuçların doğru olarak değerlendirilebilmesi için, öncelikle erken tasarım süreci ve bu süreçteki tasarımcı davranışları incelenmiştir. ConGen'in, bir program olarak ne gibi yeni özelliklerinin ve eksikliklerinin bulunduğu anlaşılabilmesi için ise bu alanda daha önce geliştirilen program örnekleri ayrıntılı olarak ele alınmıştır. Oluşturulan bu iki yönlü (tasarımcı – bilgisayar) arka plan bilgisine dayanarak, ConGen ile türetilen tasarım örnekleri değerlendirilmiştir. Elde edilen bilgiler ışığında ConGen'in ve benzeri türetici araçların kullanımlarına ve geleceklerine ilişkin sonuçlara varılmıştır.

1.1 Tezin Yapısı

“Erken Tasarım Süreci” başlıklı ikinci bölümde, tasarım sürecinin nasıl kavranabileceğine ilişkin günümüzün yaklaşımları açıklanmaktadır. Türetici sistemlerin tasarım sürecine etkisini kavrayabilmek için iki önemli sorunun karşılığı aranmaktadır: Tasarımcının, kullandığı içsel ya da dışsal araçlarla geliştirdiği ilişki, tasarım sürecini nasıl yönlendirir? Bu ilişkinin türetici sistemlerde ve genel olarak bilgisayar destekli tasarımdaki karşılığı nedir?

“Türetici Sistemler ve Tasarım” başlıklı üçüncü bölümde, türetici sistem kavramının anlamı, ne tür yazılımların türetici sistem sayılabileceği açıklanmaktadır. Bu tür sistemlerin tasarımda kullanımlarının geleceği üzerine yorum geliştirilmesine olanak sağlayacak, teorik arka plan oluşturulmuştur. Bu alanda yapılmış akademik ve akademik olmayan çalışmalardan örnekler verilmiştir. Bu örneklerin farklı özelliklerine ve uygulama alanlarına rağmen türetici sistem başlığı altında, tasarım sürecinde nasıl yer alabilecekleri tartışılmıştır.

“Bir türetici sistem denemesi: ConGen” başlıklı beşinci bölümde, daha önceki bölümlerde bahsedilen türetici sistemlerin ne gibi özelliklere sahip olabileceğini ve bu sistemlerin tasarım sürecine etkisini gözlemleyebilmek için geliştirilen, 3ds Max eklentisi (plug-in) ConGen anlatılmaktadır. Ayrıca ConGen için öngörülen kullanım metodolojisi bu bölümde açıklanmıştır.

“ConGen ile üretilmiş tasarım örnekleri” başlıklı altıncı bölümde, farklı tasarım disiplinlerine ait örnekler detaylı olarak ele alınmış, ConGen kullanımının her örnekte ne gibi faydalar sağlayabildiği tartışılmıştır.

Sonuç bölümünde ise, önceki bölümlerde verilen örnekler ve teorik arka plandan hareketle ConGen programının tasarım sürecinde daha etkin olarak kullanılabilmesi için eklenebilecek özellikler ele alınmış, türetici sistemlerin erken tasarım sürecinde kullanımlarının ne gibi faydalar sağlayabileceği ve gelecekte bu alanda yapılacak çalışmaların tasarım dünyasında ne gibi bir yere sahip olabileceği tartışılmıştır.

2. ERKEN TASARIM SÜRECİ

Tasarım yaşam-çevremizi dönüştürebilmek için temel oluşturan, insanoğlunun en karışık, en fazla zekâ gerektiren aktivitesidir. Bu alanda yapılan birçok çalışmaya rağmen hala eksiksiz bir şekilde bilinmemesi veya anlaşılammış olması şaşırtıcı bir gerçektir (Gero 1995). Tasarım sürecinin aşamaları konusunda tam bir fikir birliği bulunmamasına rağmen, McGinty'nin (McGinty 1979) önerdiği mimari tasarım süreci tanımlaması, farklı tasarım alanlarında da kullanılabilir. Önerilen süreç tanımı beş ana adımdan oluşur:

- 1- Şematik tasarım
- 2- Tasarım geliştirme
- 3- Sunum ve değerlendirme
- 4- Detaylandırma
- 5- Uygulama

Şematik tasarım aşamasında, eskizler ve çizimler aracılığıyla tasarlanacak nesnenin genel karakteristikleri belirlenir. Geçici tasarım kararları verilir. Tasarım geliştirme aşamasında, projenin özgün karakteri ve genel amacı tanımlanır. Sunum ve değerlendirme aşaması ise tasarım nesnesinin müşterisi, mal sahibi ya da jüri ile gerçekleştirilen ve ilk iki aşamayla birlikte tekrarlanan bir aşamadır. Bu aşamada tasarım kararları onaylanır. Detaylandırma aşamasında, tasarım kararlarının gerektirdiği uygulama dokümanları ve açıklamaları hazırlanır. Uygulama aşamasında ise tasarım projesi hayata geçirilir.

Bu bölümde, erken tasarım süreci deyimini, şematik tasarım ve tasarım geliştirme aşamaları için kullanılmaktadır. Üç, dört ve beşinci aşamalar geliştirilen tasarım fikirlerinin sonuçlandırılması ile ilgilidir.

Yeni bir tasarım fikri geliştirmeye çalışan kişi, hasta-tanımlı* (ill-defined) hedeflerle yola çıkar. Ancak gerçek dünya olgularından hareketle olası kullanım senaryoları üreterek buradaki problemlere, verili koşullar dâhilinde çözümler ya da alternatifler üretir. Bunun için çalışırken kendini varsayılan kullanıcının yerine koyar ve çeşitli araçlarla kullanıcının

* Hasta-tanımlı (ill-defined problem) problem (Simon 1973) , çözüm metodu problem tanımından çıkarılamayacak, doğrusal bir süreçte mantıksal kararlarla çözülemeyecek bir problemdir. Ve de tek bir doğru cevaba sahip değildir, farklı doğru cevaplardan oluşan çözüm aralığına sahiptir.

deneyimini anlamaya çalışır. Bu aşamada tasarımcının kullandığı eskiz gibi araçlar sadece görsel hafızaya yardım işlevi değil aynı zamanda kullanıcı deneyimini kavramamıza ve sentezlememize yardımcı olurlar. Bu süreçte en önemli mekanizmalar, karar verme ve seçim mekanizmalarıdır. Seçim yapabilmek için gereken bilgi ise çözüm arayışları sırasında oluşturulur.

Schön (1992) tasarım sürecini, tasarımcının kâğıt üzerine aktardığı düşünceleriyle kendisi arasındaki diyalog olarak tanımlar. Bu basit model tasarımcının, problem verilerinin sembolik temsilleriyle görsel ve zihinsel bir etkileşim kurduğunu ortaya koymaktadır. Dış temsillerden, bunlar eskizin yapıldığı kâğıt parçası ya da bilgisayar ekranı vb. olabilir, gelen geri besleme(feedback), düşüncenin ve ona bağlı aracın hareketine yol açar, bu hareket yeni dışsal temsiller kurar ve süreç bu şekilde döngüsel olarak devam eder.

Gero (1985) da tasarım sürecinin erken aşamalarını çizmek ve gözlemekten oluşan sürekli bir zincir olarak tanımlar. Tasarımcı resmin aldığı şekli görür, bazı düzeltmeler ve iyileştirmeler yapar, çeşitli değişikliklerin etkilerini dener ve yeni görsel veriler edinir. Konu edilen resim basit bir çizim değildir, tasarımcı zihnindeki yapıya şekil verir ve ortaya çıkan şekilde amaçları ve işlevleri görür. Bu aşama tasarım yaratıcılığının temelini oluşturur.

2.1 Tasarım Sürecinde Tasarımcı Davranışları ve Araçları

Tasarımcının, tasarlama sürecindeki davranışlarına, tercihlerine ilişkin birçok araştırma yapılmış, yapılmaktadır. Ünlü tasarımcılar kendi tasarım yöntemlerini açıklayan yazılar yayınlamış ve bu yayınlar tasarımcı olmak isteyen insanlar için çok faydalı olmuşlardır. Ancak bu örneklerdeki yöntem ve tercihler o denli kişiseldirler ki genel kabul görmüş bilimsel bir teorinin oluşabilmesi için yetersiz kalırlar.

Candy ve Edmonds (1997) tasarım sürecinde tasarımcıların paylaştıkları genel karakteristikleri şöyle sıralarlar:

- Tasarımın anlamsal ya da işlevsel açıdan özel öneme sahip bölümleri öncelikli olarak ele alınır.
- Tasarımcılar tasarım süreci sırasında daha önce belirledikleri hedeflerini değiştirip yeni hedefler ve sınırlamalar koyarlar. Bu döngü tasarım sonuçlanıncaya kadar devam eder.
- Hedeflerin ve sınırlamaların döngüsel olarak yeniden tanımlanması, tasarımcının

hasta-tanımlı problemlerle uğraşırken geçirdiği öğrenme sürecinin bir parçasıdır.

- Tasarımcılar çözüm üretebilmek için geçici tasarım sınırlamaları belirlerler, bu sınırlamalar ana önerinin yanlış bir şekilde kurulmasına neden olsa da hedefler ve sınırlamalarla oynama özgürlüğü, radikal fikirlerin ortaya çıkmasında belirleyici olur.
- Tasarımcılar genellikle karşılaştıkları problemleri hasta tanımlı problemler olarak görmeye eğilimlidirler, bu problemler düzgün tanımlanabilir olsalar bile. Bu belirli araştırmaların tasarım problemlerini tanımlama çabalarının tersine gelişir. Anlaşılan tasarımcılar hasta tanımlı problemlerin sağladığı esnekliği tercih etmektedirler.

Örnekleri verilen tasarımcı davranışları, bilgisayar programlarının tasarım geliştirmede yeterli şekilde kullanılmamalarının ana nedenleridirler. Tasarımcının düşünceleri çoğu zaman sezgisel hatta irrasyoneldir ve nadiren tekrarlanabilir kalıplara uyar. Hesaplama (computing) ise tahmin edilebilirlik gerektirir. Hesaplama katı bir algoritmik süreç iken insan düşüncesi her zaman tahmin edilemez. Bu uyuşmaz yapıları nedeniyle bir bilgisayardan, insan yaratıcı sürecini tam anlamıyla taklit edebilmesi beklenemez. Ancak, çeşitli bağlamsal parametrelerin yardımıyla tasarımcının ihtiyaç duyacağı form alternatifleri üretebilecek algoritmalar geliştirilebilir (Gardner 1997).

2.1.1 Eskiz'in Tasarım Geliştirmedeki Kullanımı

Goldschmidt'e göre (1991), tasarım eskizleri içlerinde birçok yönelimi barındıran gevşek ve belirsiz çizimlerden oluşurken, kesin çizimler çoğunlukla yardımcı araçlar kullanarak gerçekleştirilen, farklı yönelimler içermeyen keskin çizgilerden oluşurlar. Tasarımcılar eskizleri çoğunlukla kendileri için, kesin çizimleri ise tasarımlarını başkalarıyla paylaşmak için yaparlar. Erken tasarım aşamalarında bu belirsiz çizim tekniğinin kullanılmasının sebebi, bu tür çizimlerin tasarım-problemi-uzayını belirli bir kesin çözümü ima etmeden yansıtmaıdır (Goel 1995). Tasarımcılar, yeni tasarım fikirleri bulabilmek için zihinlerinde canlandırdıkları ya da daha önce gördükleri imgeleri kullanarak çizimler yaparlar. Bu çizimleri birbirleriyle ilişkilendirerek çözümlerini geliştirirler. Bu süreç tasarımcının aklındaki imge ortaya çıkıncaya kadar sürer. Bu döngüsel süreç sonrasında tasarımcı yeni bir fikir bulmuş ya da geliştirmiş olur. (Kwon D. Y. 2003)

İnsan beyni eşzamanlı olarak veri depolama ve bunlara zihinsel operasyonlar yapma konusunda sınırlıdır. Bu sınırlamalar hafızanın eskiz gibi ortamlara aktarılmasıyla hafifletilir. Böylelikle zihinde oluşturulan tasarım kavramlarına ilişkin özellikler tekrar tekrar

incelenebilir ve geliştirilebilirler. Çizgi romanlar, tasarım eskizleri ya da yol haritaları gibi bazı özel çizim türleri, resim ya da görsel algılamada yer alamayacak ek enformasyonlar içerirler (yön göstermek için oklar, sıcaklığın yükseldiğini anlatabilmek için çarpık çizgiler ya da kelimelerin şekilleriyle anlatılan ünlemler vb.). Özetle, bu çizimler insanın algıladıklarını değil o şey hakkındaki kavrayışlarını açığa vururlar(Twersky 1999).

Çizim sırasındaki ilk adım kâğıt üzerinde nereden başlanacağına karar vermektir. Bu karar programdaki "move" işlemine karşılık gelir. İkinci karar, kalemin hangi yönde hareket ettirileceğidir. Bu karar da öklidyen transformasyonların karşılığıdır "rotate&reflect". Üçüncü karar çizginin ne kadar uzunlukta sonlanacağını belirlemektir ki bu da boyut transformasyonlarına eşittir. Bu adımların hepsi zihinsel analogi problemlerinde kullanılan sırayla işlemektedir. Çizimin yapılma düzeni, zihinsel transformasyonların düzenini tekrarlar. Zihinsel düzen çizim düzenini içselleştirmiştir. Çizimin üretilmesindeki düzen tasarımın arkasındaki konsept kurulumuna dair ipuçları içerir.

Bilgisayarların yalnızca kesin çizimler üretmekte değil, eskiz gibi çoklu kavrayışlara imkân verebilecek çizimler de üretmekte de kullanılabileceği açıktır. Ancak tasarımcının bu araçları kullanımını kolaylaştıracak bazı ek araçlara ihtiyacı vardır (bilgisayar ekranına elle çizimi destekleyen kalemler ve bunlara bağlı sistemler gibi). Gerekli yardımcı araçlar ve bunların kullanımını kolaylaştıracak yazılımlar geliştirildiği takdirde, eskizin, insanın evrimsel gelişiminde kendiyle-iletişim eyleminin detaylı bir dışavurumu olduğunu unutmadan, bilgisayar programlarının, eskiz araçlarının, mantıklı, tatmin edici ve yüksek kaliteli bir eşdeğeri olduğu kabul edilebilir.

2.1.2 Tasarım Sürecinde Analogi

Analojik düşünme genel anlamıyla iyi bilinen bir duruma (kaynak) ait bir bilginin, kavranmaya ya da geliştirilmeye çalışılan bir durumda (hedef) kullanılmasıdır (Novick 1988). Gentner (1983) gibi teorisyenlere göre, analogiler, yüzeysel ya da strüktürel analogiler olmak üzere iki kategoride ele alınabilirler. Yüzeysel analogilerin oluşturulması daha kolay iken, strüktürel analogilerin kurulabilmesi için kaynak ve hedef durumlar arasında strüktürel anlamda derin bağlar bulunması gerekir. Yüzeysel analogiler çoğunlukla aynı alandan seçilmiş örnekler arasında kurulurlar (Dejong 1989), farklı alanlardan seçilmiş örnekler arasında analogi kurulabilmesi ancak strüktürel bağlantıların bulunabilmesi ile mümkün olur.

Tasarım sürecinde, görsel referansların, analogilerin ve metaforların kullanımına erken tasarım sürecinde rastlanılabilir. Kullanılan analogiler genellikle strüktürel analogilerdir. Tasarımcının

referans formlar ararken kullanabileceği tüm yöntemlerin belirlenmesi imkânsız olsa da iki ana yöntemden bahsedilebilir (Do ve Gross 1995). İlk olarak, tasarımcılar kendi zihinlerindeki biçime benzer biçimler ararken, bunları kâğıda aktarırlar ve o çizim bir tasarım fikrini tetikler. İkinci yöntem ise, tasarımcıların eldeki tasarımla bir kavram aracılığı ile bağlantısı olan formları keşfetmelerdir (Örn: Sidney Opera Binası). Bu aşamada tasarımcı yapısal ya da fonksiyonel problemlerle ilgilenmekten çok tasarım konusu için, form ya da kavramsal nitelik etütleri yapmaktadır. Bu konularda yapılan tercihler belirli bir olgunluğa geldiğinde diğer sorunlar gündeme alınacak ve fiziksel form bunların ışığında yeniden değerlendirilecektir (Do ve Gross 1995).

Tasarım problemi büyük oranda probleme yüklenmiş kısıtlarla yönetilir. Tasarımcının bir tasarım problemini çözebilmek için çok sayıda kısıtlamayı devreye sokması sık rastlanılan bir olgudur. Kısıtlamaların eklenmesi, belirlenmiş prensipler, özgün özellikler ya da analogi gibi bilişsel süreçlerin rehberliğinde gerçekleşir (Eckert 1999). Bu kısıtlamalar olası tasarım çözümleri uzayının daraltılması için kullanılır.

Bilgisayar programları, henüz tasarımcının strüktürel analogiler kurma yeteneğinin çok uzağındadırlar. Ancak bilgisayarlar insana göre neredeyse sınırsız hafıza ve listeleme becerisine sahiptirler. Ve bu özellikleri sayesinde tasarım analogileri oluşturmakta büyük bir yardımcı olarak kullanılabilirler.

2.1.3 Tasarım Sürecinde Şekil Belirmesi (Shape Emergence)

Tasarım sürecinde şekil belirmesi, varolan bir tasarımda daha önce görülemeyen yeni bir fikrin ya da özelliğin algılanması olarak tanımlanmaktadır (Cross 1997). Birçok tasarım probleminde çözümler çeşitli seçenekler olarak yer alırlar, ancak tasarımcı onları çözüm olarak seçemediği ya da göremediği sürece ortaya çıkamayacaklardır.

İlk bakışta belirlenmeyen, tasarım probleminin içeriğinde bulunmayan çeşitli özellikleri, karakteristikleri tanıyabilme becerisi çok önemli bir insan yeteneğidir. Bu tür görsel gruplamaların, tanımlamaların zihinde nasıl gerçekleştiği ile ilgili araştırmalar insan düşünce süreçlerinin ve algılayışının anlaşılabilmesi açısından önemli bir yere sahiptir.

Stiny'nin (Stiny 2001) şekil belirmesi hakkında matematiksel görüş açısına göre yaptığı çalışma, belirsizliğin, şekillerin en temel özelliklerinden kaynaklandığını gösterir. Şekiller tanımlı alt parçalara sahip değildir. Bu nedenle birçok farklı parçaya herhangi bir yolla bölünebilirler. Bir şeklin içinde her zaman yeni şeyler görülebilir. Bir şeklin yapılış biçimi ile

bölünme biçimi birbirinden tamamen bağımsızdır. Ancak anlamsal şekil belirmeleri, yeni bir anlamsal içerik tarafından yönlendirilmiş, ya da tetiklenmiş olmalıdırlar. Suwa (1999), bir mimarın bilişsel süreçlerini gözlemlemiş ve yeni bir tasarım ihtiyacı ya da yeni bir kavram ortaya çıktığında tasarımcının bazı özellikleri çok daha farklı bir perspektiften ele almaya başladığını ve bunun onu yeni buluşlara götürebildiğini kaydetmiştir.

Bir şeklin içindeki alt-şekilleri görme yeteneği her insanda bulunan, görsel ve bilişsel bir beceridir. Yine de insan, çaba göstermeksizin görmek yerine, bu alt şekilleri aramak zorundadır ve bazı şekilleri fark etmek gerçekten çok zordur. Bu da görsel algı işleyişinin bilişsel içeriği olmadan ele alınamayacağını kanıtlar. Görme üzerine çalışan birçok psikolog, örneğin Arnheim (1969) algısal tanımlamanın görsel prototiplerin rehberliğinde gerçekleştiğinden bahseder. Bu görüş görsel algılamayı harekete geçiren şeyin yorumlamayı beraberinde getiren prototip sınıflarının olduğunu varsayar. Arnheim'in görsel prototiplerini, Stiny'nin (2001) şekil sınıfları olarak yorumlarsak bu iki teorinin birbiriyle çelişmediğini görebiliriz (Oxman 2002).

Soufi ve Edmonds (1996) görsel belirme sürecini iki adımda açıklarlar:

- Görsel kalıpların tanımlandığı algısal yorumlama süreci
- Bu görsel kalıpların yeni yapıların üretilmesi için kullanılması.

Bu şekilde tasarımcının varolan görsel hafızası şekil manipülasyonuna, transformasyonlarına ve yenilerinin keşfine rehberlik eder. Mimarlık gibi düzenlenebilir şekilsel kalıplar ile ilgilenen disiplinlerde, imgeler, karmaşık anlamsal içerikleri sembolize etmeye yarayan görsel hafıza araçları olarak çalışırlar. Bu şekiller hakkında akıl yürütebiliriz çünkü onları yorumlayabiliriz. Görüntüler yorumu desteklerler ve akıl yürütmeye yol açarlar (Oxman 2002).

Bilgisayarların şekil belirmelerini algılamaları ve kullanıcılarına sunmaları henüz yalnızca çok iyi tanımlanmış problem alanlarında ve sınırlandırılmış belirme kategorileriyle (görsel simetri, görsel hareket, görsel denge (Gero 1995) vb.) mümkün olmaktadır. Bu alanda geliştirilen çoğu üretici sistem uygulaması, beklenmedik şekil üretimleri ve bunların değiştirilebilir temsilleri aracılığıyla, kullanıcının şekil algılayışını değiştirerek, şekil belirmeleri konusunda tasarımcıyı desteklemeyi amaçlamaktadırlar.

3. TÜRETİCİ SİSTEMLER ve TASARIM

Türetici sistem terimi geniş anlamda, bir girdi setini temel alarak tasarım türetebilen tüm metotlar için kullanılmaktadır. Bu türetici metotlar, genetik algoritmaları, biçim gramerleri gibi kurallı yeniden yazma yöntemlerini ve heuristic yöntemleri içerir. (Eckert, Kelly ve Stacey 1999)

Türetici sistemler basit hesaplama kavramı üzerine inşa edilirler. Sisteme girilen veriler belirlenmiş fonksiyonlara göre dönüştürülmesi ya da ilişkilendirilmesi ile belirli sonuçlar elde edilir. Bir tasarımcı belirli şekiller girdiğinde, bilgisayar, tasarımcının belirlediği algoritmaları kullanarak olası şekil kompozisyonu oluşturur. Tasarımcı daha sonra bu şekil yapılandırmalarını kullanarak türetme işlemine devam edebilir. (Kwon 2003)

Tasarımcının yönetimindeki interaktif araçlar olarak, türetici sistemler, tasarımın biçimsel özelliklerini ve yapısal problemlerin çözümünü geliştirmeye yönelik tasarımcı yeteneklerini ortaya çıkarabilir ve destekleyebilir. Bir türetici sistem, aşağıdakilerle tutarlı olan tüm tasarım alternatiflerini geliştirebilir:

- Tasarım işini tanımlayan girdiler.
- Türetici kurallar ve algoritmalar.
- Temsil formalizmi içinde kurulmuş olan, tüm tasarım uzayını kapsamaya olanak verecek sınırlamalar.

Tasarım alternatifleri uzayı kontrol edilemeyecek kadar büyükse daha ileri sınırlamalar getirilebilir ve bunlar türetici kurallar ya da algoritmalar içerisine kurulabilirler. Ya da türetme yöntemleri kademeli olarak çalıştırılabilen, her kademede tasarımcının seçimi ile diğer kademeye geçebilen sistemler haline getirilerek kontrol sağlanabilir. Başlangıçta belirlenen türetici kurallar, sistemin neyi üretip üretmeyeceğini, ya da üretip eleyeceğini belirler. Tasarım için üretilen türetici sistemler problem belirlenmesi, tasarım üretimi ve tasarım geliştirmesinden oluşan bir döngü içinde hareket ederler. Bu da insan tasarım alışkanlıklarına çok benzer bir süreç oluşturur.

Tamamlanmış tasarımların oluşturulmasında ve bunların geliştirilmesinde insanları desteklemekte türetici sistemlerin sundukları birçok avantaj vardır(Eckert, Kelly ve Stacey 1999):

- Bir defada çok sayıda alternatif üretilebilir ve belirlen problem için tüm çözüm uzayı

incelenebilir.

- Yeni tasarımların üretilme süresi geleneksel yöntemlere göre çok kısadır.
- Yeni tasarımlar tamamen hesaplama yöntemleriyle ya da tasarımcının uygulamakta zorlanacağı tasarım kriterlerinin bilgisayar tarafından kontrol edilmesiyle üretilebilir.
- Tüm tasarımlar istenilen düzeyde teknik detaya, anlaşılabilirliğe, bitmişliğe ya da soyutluğa sahip olabilirler.
- Tüm tasarımlar kullanıcının isteği doğrultusunda, resimler, diyagramlar ya da tercih edilen herhangi bir başka formda sunulabilirler.
- Tüm tasarımların yapısal özellikleri, program içerisindeki tanımlamalarının kurulumu aşamasında hassas ve hiçbir belirsizliğe yer vermeyecek şekilde belirlenmiş olurlar.

Ayrıca biçim üretme algoritmaları tasarımcıların kompleks biçimsel kalıplar oluşturmaları ya da eldeki biçimleri dönüştürmek için kullanılabilirler. Dahası bu algoritmalar "tasarım bilgi sistemleri" (design knowledge systems) ile ilişkilendirilerek bilgisayarın biçim üretimini ya da manüplasyonunu kontrol etmesi sağlanabilir.(Kwon 2003)

3.1 Türetici Sistem Örnekleri

Türetici tasarım programları ve teknolojilerinin geliştirilmesi, günümüzün kabul görmüş önemli araştırma alanlarından biri haline gelmiştir. Türetici sistem araştırmalarına ilgi gösteren üniversiteler, yeni geliştirilen programlar ve bunların uygulama örnekleriyle, tasarım ve sanat çevrelerinin dikkatini bu konuya çekmektedirler. Türetici sistem deyimi bu alandaki birçok farklı çalışmayı kapsamaktadır. Ele alınan örnekler, tümüyle bağımsız olmayan ancak temel özellikleriyle birbirlerinden ayırt edilebilecek türetici sistemlerden seçilmişlerdir.

3.1.1 Parametrik Tasarım Araçları

Parametrik tasarım araçları, günümüzün tasarım programlarında da sıklıkla kullanılmaya başlanılan, temel türetici araçlardır. Bu araçlar, tasarımcının bir tasarım ailesini, belirlenmiş anahtar veriler ya da parametrelerle tanımlamasına olanak sağlarlar (Gross 1989).

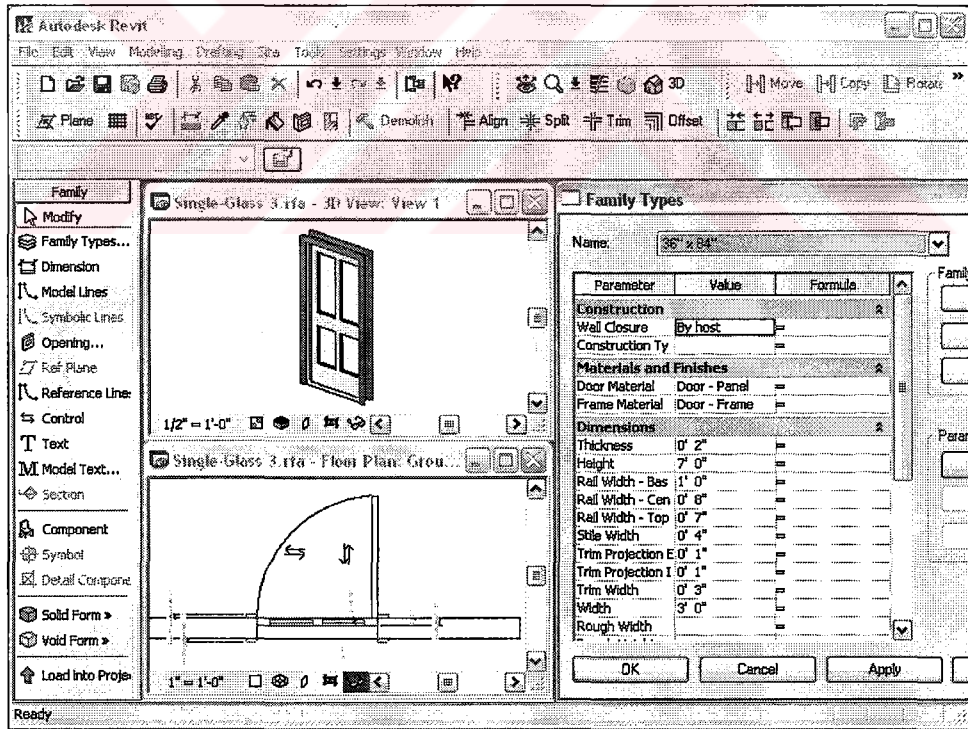
Örneğin, bir merdiven, katlar arasındaki yükseklik ve basamak sayısıyla tanımlanabilir. Parametrik tasarım aracı, bu tanımlamadan hareketle, merdiveni oluşturabilecektir. Türetimi mümkün kılan, program içerisindeki parametreler hiyerarşisini yöneten yönlendirmelerdir.

Örnekte verilen merdiven için, kat yüksekliği ve basamak sayısı, yönlendirici, rıht yüksekliği, basamak derinliği, total merdiven uzunluğu ise yönlendirilen parametrelerdir. Birçok ticari BDT (Bilgisayar Destekli Tasarım) yazılımı, bu tip parametrik tasarımlara izin vermektedirler. Ancak, en gelişmiş ticari parametrik tasarım aracı Autodesk firmasının ürettiği Revit programıdır.

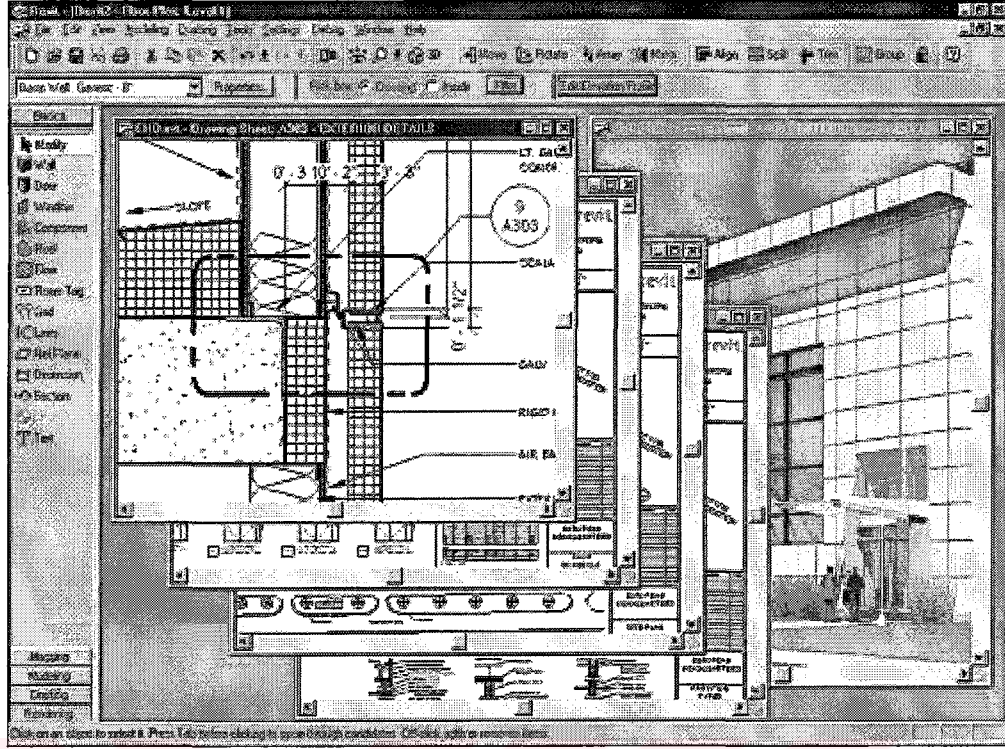
3.1.1.1 Bir parametrik mimari tasarım aracı: Revit

Revit ilk ticari parametrik yapı modelleme programıdır. Revit'in parametrik yapı modelleyicisi, akıllı bina bileşenlerinden, görünüşlerinden ve bunların açıklamalarından oluşur. Tüm bu bileşenler hem parametrikler hem de Revit'in yüksek performanslı değişim motoru sayesinde ilişkilendirilmişlerdir. Bu sayede, yapılan herhangi bir tasarım değişikliği projenin tüm dokümantasyon ailesine yansıtılabilir.

Örneğin, parametrik bir duvar diğer yapı elemanlarıyla ilişkisini anlar. O duvar üzerinde bulunan çatının eğiminde bir değişiklik yapıldığında, duvar da o eğime uyum gösterecek şekilde değiştirilir. Ve bu değişiklik o duvarla ilgili tüm dokümanlara yansıtılır.



Şekil 3-1 Parametrik nesne kütüphanesi arayüzü (Revit web sitesi)



Şekil 3-2 Proje dokümanlarının aynı arayüzde kullanımı (Revit web sitesi)

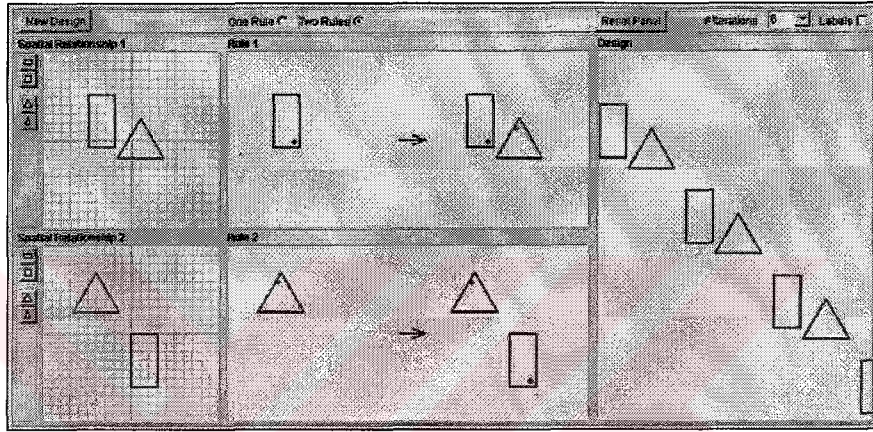
3.1.2 Biçim Gramerlerini Kullanan Türetici Sistemler

Biçim gramerleri, yetmişli yılların sonunda, görsel kompozisyonların, biçimsel bir dille anlatılabilmesini sağlamak amacıyla Stiny ve Gips tarafından icat edilmiştir (Stiny, 1976). Bir biçim grameri, en yalın ifadeyle, bir şeklin ya da şekil parçasının başka bir şekille nasıl değiştirilebileceğini belirleyen kurallar ailesinden oluşur.

Stiny tarafından yazılan, “Two exercises in formal composition” (Biçimsel kompozisyonlar üzerine iki alıştırma), adlı makalede, biçim gramerlerinin nasıl kullanılabilceğine dair iki alıştırma sunulmuştur. Birinci alıştırma, biçim gramerlerinin, yeni tasarım üretmek için nasıl kullanılabilceğini, ikinci alıştırma ise biçim gramerlerinin, varolan ya da bilinen bir tasarım dilini analiz etmek için nasıl kullanılabilceğini göstermektedir (Knight, 1999). Biçim gramerleriyle ilgili yapılan çalışmaların büyük çoğunluğu bu iki yöntemden yola çıkmışlardır. Özellikle erken tarihli, “The Palladian Grammar” (Stiny ve Mitchell, 1978), “The language of the prairie: Frank Lloyd Wright’s prairie houses” (Koning ve Eisenberg 1981) gibi çalışmalar, birinci yöntemi izleyerek geliştirilmiş detaylı analiz çalışmalarıdır. Daha yakın tarihli çalışmalar ise, biçim gramerleriyle yeni tasarımlar üretmek ya da varolan bir stilin çeşitlemeleri üretebilmek için kullanılmaktadır. Aşağıda, son on yılda bu alanda yapılan çalışmalardan örnekler verilmiştir.

3.1.2.1 İki Boyutlu Biçim Türetme Aracı: Shaper 2D (McGill, 2001)

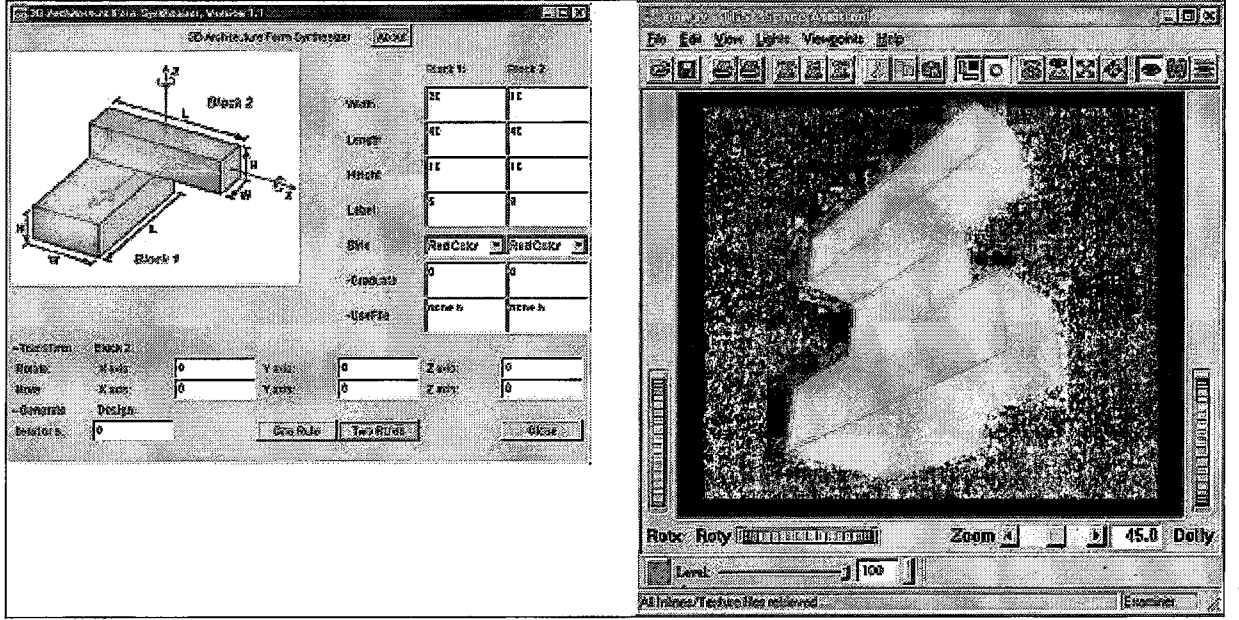
Bu program aracılığıyla, farklı noktalarından etiketlenecek iki şekil (kare, dikdörtgen, eşkenar ve ikizkenar üçgen) arasındaki uzamsal ilişki kuralından türetilen şekiller incelenebilir. Tüm şekiller, fare (mouse) yardımıyla istenilen şekilde boyutsal ve öklidyen transformasyonlara uğratılabilir. Yapılan transformasyonların etkisini, eşzamanlı olarak gözleyebilmek mümkündür. Ancak sayısal değer girme seçeneği sunulmamıştır.



Şekil 3-3 Shaper 2D programının arayüzü (Knight 1999)

3.1.2.2 Üç Boyutlu Biçim Türetme Aracı: 3D Shaper (Wang ve Duarte 2001)

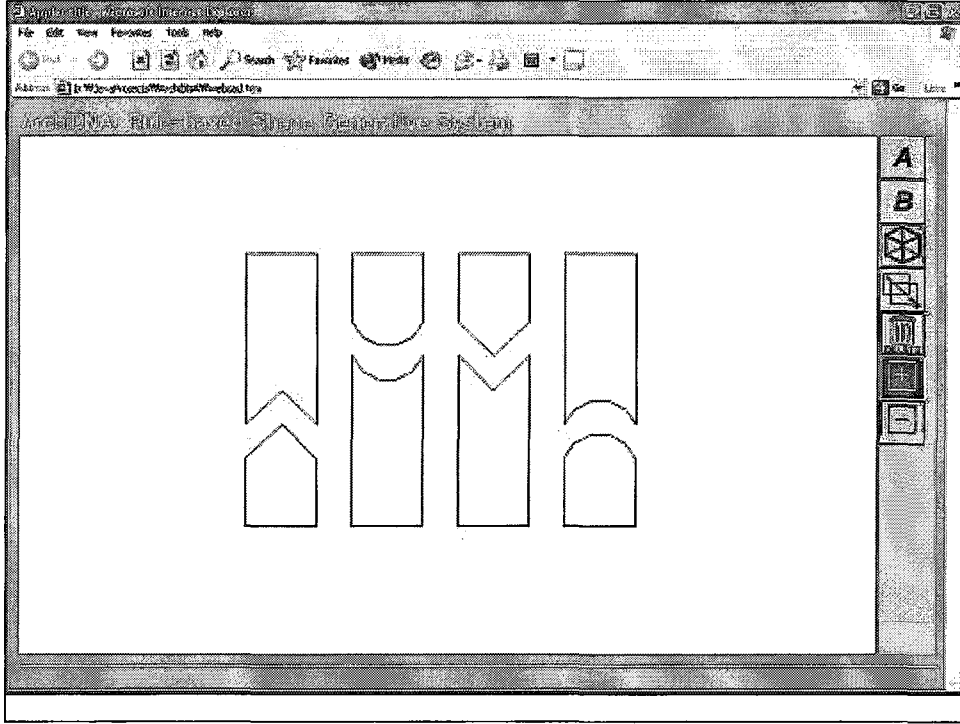
Program aracılığıyla, küp, sütun, çokgen gibi düzgün geometrik şekiller arasında tanımlanan uzamsal ilişkiden türetililecek tasarımlar incelenebilir. Program arayüzü, iki parçalıdır. Bunlar, iki şekil arasındaki ilişkinin sayısal olarak tanımlandığı panel ile türetilen şekillerin incelenebildiği üç boyutlu grafik panelidir.



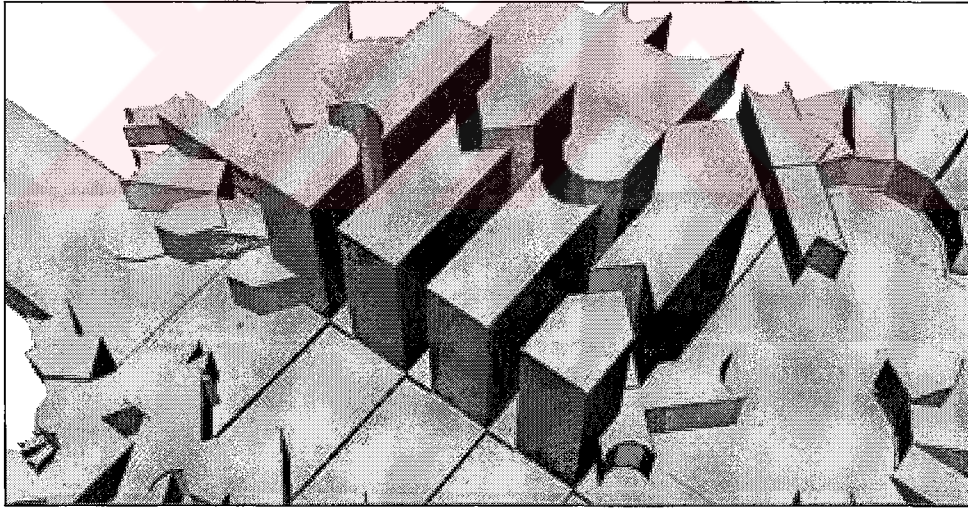
Şekil 3-4 3D Shaper programının arayüzü (Knight 1999)

3.1.2.3 Biçim Gramerleriyle Stil Türetimi: Archi-Dna (Kwon 2003)

Bu sistem, varolan bir mimarlık stilini, bilgisayar dilinde programlayıp, bu stile ait yeni tasarımlar üretebilmeyi sağlamaktadır. Seçilmiş olan mimari dil, Peter Eisenmann'ın Frankfurt Almanya'da yapılan Biocentrum projesinden alınmıştır. Eisenmann bu yapıda DNA'da bulunan farklı aminoasitleri simgelemek için kullanılan biçimlerden ve aminoasitlerin dizilim kurallarından esinlenmiştir. Archi-Dna, Eisenmann'ın kullandığı dizilim ve transformasyon kurallarına uyararak, yeni tasarımlar türetir. Kullanıcının, kendi geliştirdiği şekillerin de program içerisine kullanılabilmesi için Archi-Dna++ adlı ikinci bir program versiyonu da geliştirilmiştir.



Şekil 3-5 Archi-Dna programının arayüzü. (Kwon 2003)



Şekil 3-6 Archi-Dna programıyla üretilmiş kentsel tasarım örneği (Kwon 2003)

3.1.3 Evrimsel Algoritmalarla Dayalı Türetici Sistem Örnekleri

Yetmişli yıllarda genetik algoritmalarının ortaya çıkışıyla, evrimsel tasarım (evolutionary design) ve ilgili yaklaşımlar, çağdaş tasarım teorisinin büyük akımlarından biri haline geldiler (Fischer ve Fischer 2003). Genetik algoritmaları, tipik olarak, bir optimizasyon probleminin çözüm adaylarının (bu alandaki çalışmalarda bireyler olarak adlandırılan) temsillerinin (bu alandaki çalışmalarda kromozom olarak adlandırılırlar) bulunduğu bir bilgisayar simülasyonu

olarak kurulurlar. Evrim, tamamen rastgele bireylerden oluşan bir topluluktan, belirli kriterlere göre seçilmiş bireylerin bir sonraki nesle geçmesini sağlayarak gerçekleşir. Her yeni nesilde, topluluğun özellikleri iyileştirilmiş olur. Seçilen bireylerin, özellikleri değiştirilerek (mutasyon ya da rekombinasyon yoluyla) istenilen uygunluk koşulları sağlanmaya çalışılır. Genetik algoritmaları tasarım sürecinde genellikle optimizasyon problemlerinde ve form üretiminde kullanılmaktadırlar. Bu bölümde, bu tür araçlara iki örnek verilecektir.

3.1.3.1 Bağımsız Bir Evrimsel Sistem: Argenia

Milan Politeknik Üniversitesi Mimarlık ve Planlama Bölümü'nden Prof. Celestino Soddu, Avrupa'da evrimsel tasarım alanının en etkili isimlerinden biridir. Halen yöneticiliğini yaptığı Milan Politeknik'te bulunan Türetici Tasarım Laboratuvarı'nda (Generative Design Lab) türetici sistemlerle ilgili çalışmalar yürütmektedir.

Soddu'nun geliştirdiği Basilica ve Argenia adlı türetici sistemler iki temel düzenekten oluşmaktadırlar. Türetim yapılacak alanın (mimarlık, endüstri ürünleri tasarımı, sanatsal çalışmalar) özelliklerine göre şekillendirilebilen bir türetici kod ve ortaya çıkan ürünün gelişimini kontrol etmeye yarayan araçlar. Soddu'ya (Soddu 2002) göre, programların temelinde bulunan kod bölümü, ürünün DNA'sı işlevini görür ve oluşturulmasında rasgeleliğe yer yoktur. Ancak bu kodun türetilerek gelişmesini sağlayan araçlar rasgeleliği de kullanarak temel kodun öngörülemeyen potansiyellerini ortaya çıkarmaya yararlar.



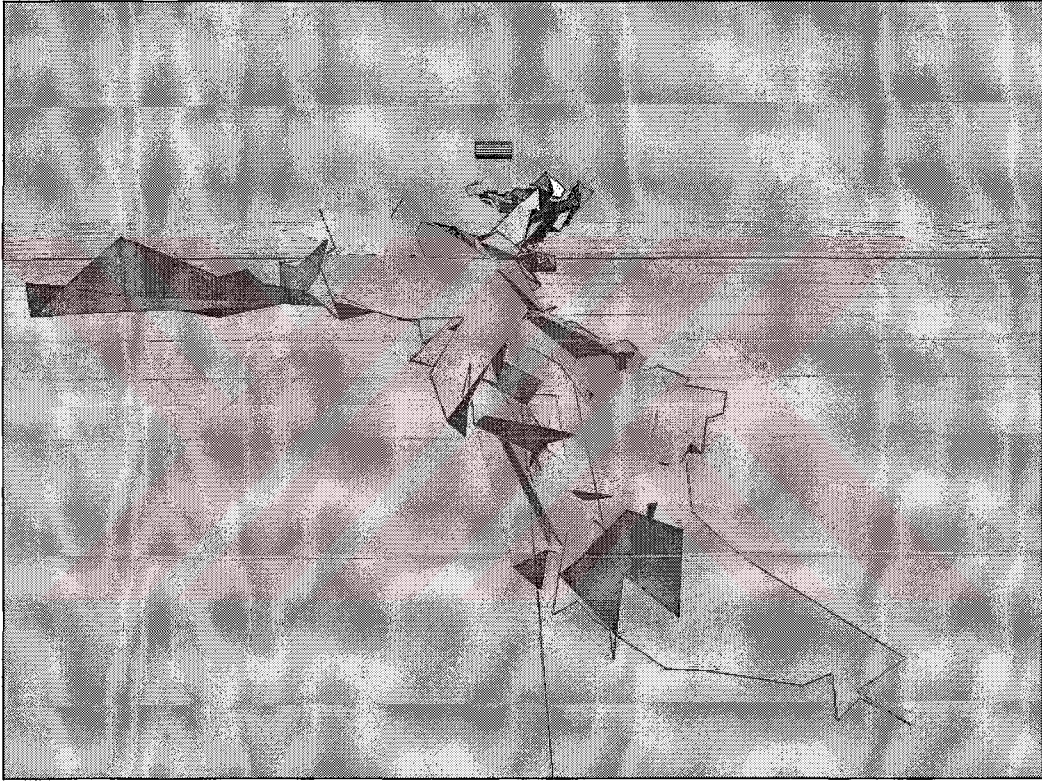
Şekil 3-7 Milano'da yapılacak yeni Fütürizm Müzesi için mevcut durum ve Argenia programıyla üretilmiş üç alternatif (Soddu 2004)



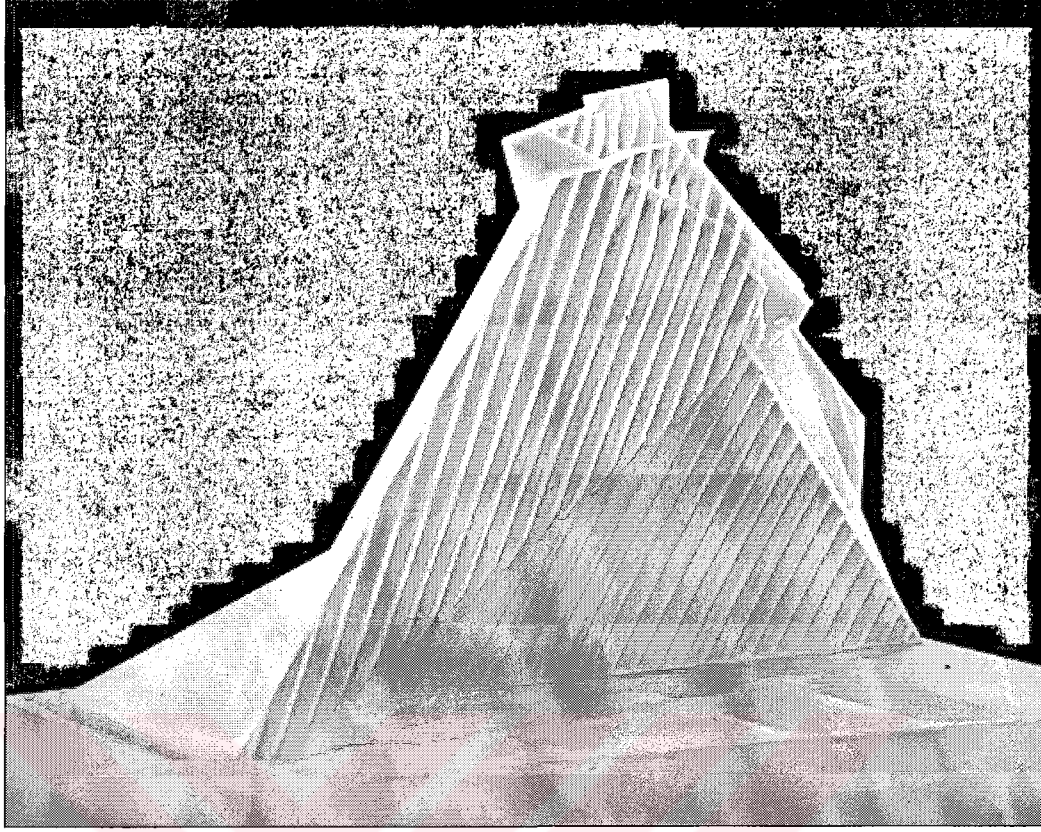
Şekil 3-8 New York kenti kimliğinden elde edilmiş kodla türetilmiş bir kent parçası (Soddu 2004)

3.1.3.2 Evrimsel Bir Eklenti (Plug-in) Örneđi: Genr8

Genr8, MIT bünyesinde çalışan araştırma grubu “Emergent Design Group”, tarafından geliştirilmiş, bir dijital yüzey üreticisidir. Program, Alias firmasının ürettiđi, üç boyutlu modelleme ve sunum programı olan, Maya ile çalışmaktadır. Genr8 programında yüzey üretimi, kapalı bir çokgenin tanımlanması ile başlar. Bu çokgen, eşzamanlı, çok boyutlu kurallarla türetilir. Türetim sırasında, yüzeyin şekli, kullanıcının tanımladıđı çevresel özelliklerin (sınırlar, çekim alanları, itici güçler, vb.) fiziksel simülasyonlarının etkisi altındadır. Bu simule edilmiş fiziksel ve tropik etkiler, organik görünümlü bir yüzeyin üretilmesini sağlarlar (Hemberg ve O'Reilly 2004).



Şekil 3-9 Genr8 tarafından üretilmiş dijital bir yüzey (kaynak: emergent design group web sitesi)



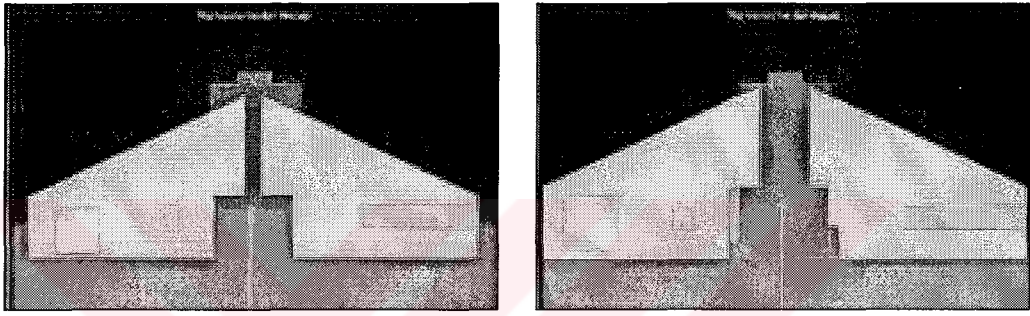
Şekil 3-10 Bir CNC makinesi aracılığıyla üretilmiş, Genr8 yüzeyi (Hemberg ve O'Reilly 2004)

3.1.4 Kısıtlama Tabanlı (Constraint-Based) Tasarım Sistemleri

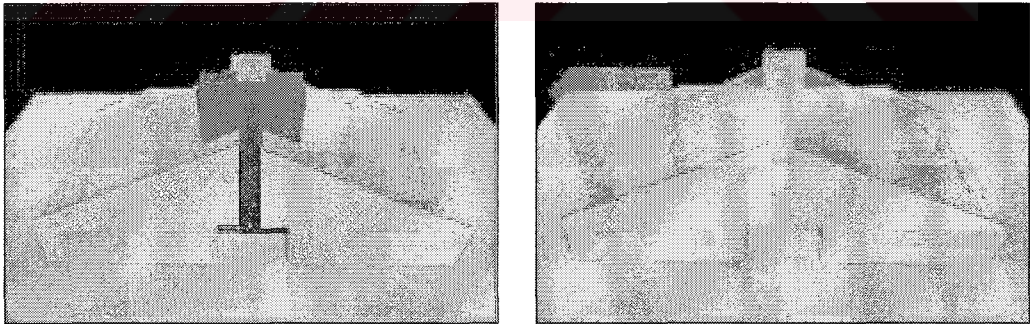
Tasarım kısıtları, tasarım probleminin bağlamını oluşturan en önemli verilerdir. Tasarımcının, dikkat etmesi gereken birbirleriyle bağlantılı gerekliliklerdir. Çeşitli nesnelere arasındaki ilişkileri düzenleyen kısıtlamaları kontrol edebilecek bir araca sahip olduğu takdirde tasarım probleminin bir yönü çözülmüş olacaktır. Bu araç sayesinde tasarımcı hem kısıtlamaları yönetebilecek hem de nesnelere ve nesnelere ilişkin kısıtlamaları sunabilecektir (Eggink 2001). Örneğin, dikdörtgen bir oda için konulan alan kısıtlaması, bu odanın birbirine dik iki duvarının boyutlarının oranını belirler. Kısıtlama tabanlı üretici sistemler, kısıtlamaların kontrolünün zor olduğu tasarım problemlerinde, bu kısıtlara uyan tasarım alternatiflerinin türetilmesini sağlarlar. Bu tür sistemlerin çalışma mantığı, parametrik tasarım araçlarının çalışma mantığına benzer ancak sistem içinde varsayılan bir yönlendirme ve parametre hiyerarşisi yoktur. Yönlendirme kullanıcının belirlediği kısıtlar sayesinde kurulmuş olur, bu nedenle tasarım sürecinin erken aşamalarında kullanılmaya daha uygundur.

3.1.4.1 Sanal Gerçeklikte Türetim: Smart Objects

Smart Objects (Akıllı Nesnelere) programı, kısıt-tabanlı üç boyutlu bir üretici tasarım aracıdır. Programda, tasarım kısıtlamaları, geçici bir çözüm oluşturan nesne modelleriyle ilişkilendirilirler. Geliştirilen program, üç boyutlu sanal gerçeklik modelleme dili (3D Virtual Reality Modeling Language-VRML) ile oluşturulmuş bir görüntüleyici ve buna bağlı olarak çalışan Java dilinde yazılmış bir yönlendirici sistemden oluşur. Smart Objects programında, tasarımcı kısıtları yerine getirdiğinde ya da ihlal ettiğinde, sahnedeki nesnelere bunu yansıtacak şekilde davranırlar. Bu şekilde, program tasarım prensiplerinin projede yer alan tüm taraflar arasındaki paylaşımına da katkıda bulunmuş olur.



Şekil 3-11 Simetri kısıtlaması konulan cephede, ana aksın bir yanındaki duvarın hareket ettirilmesi, aksın diğer yanındaki duvarın pozisyonunu değiştirmektedir (Eggink, Gross, Do 2001).

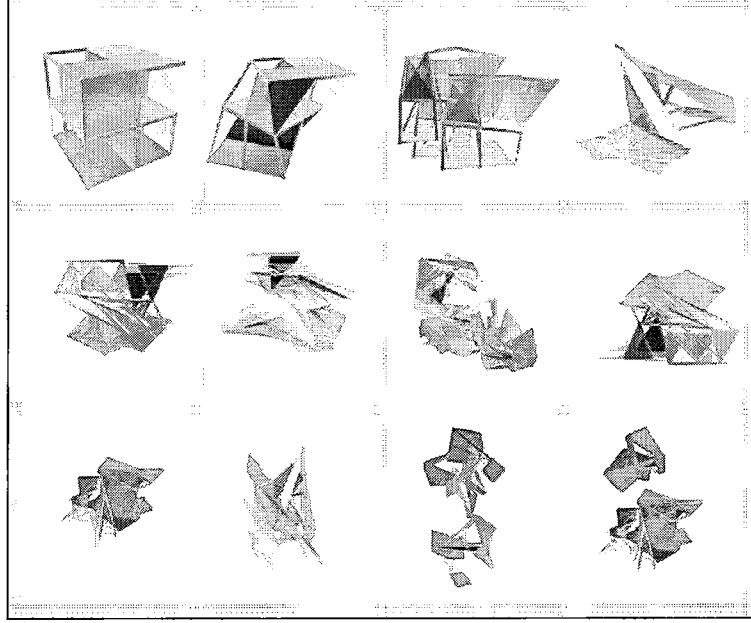


Şekil 3-12 Şömine kutusu, merkezdeki akstan uzaklaştıkça saydamlaşmaktadır (Eggink, Gross, Do 2001).

3.1.4.2 Bir 3ds Max Eklentisi: Xp-Gen

Xp-Gen (Pak 2003), 3ds Max programına eklenen bir üretici araçtır. Bu araç yardımıyla, kullanıcı seçilen nesnelere üzerinde, program içerisinde bulunan max değiştiricilerinin etkilerini test edebilir. Kullanıcının belirlediği maksimum limitlere uyacak şekilde pozisyonel ve öklidyen transformasyonlar rastgele uygulanarak, yeni tasarım fikirleri doğurabilecek

alternatifler üretilir.

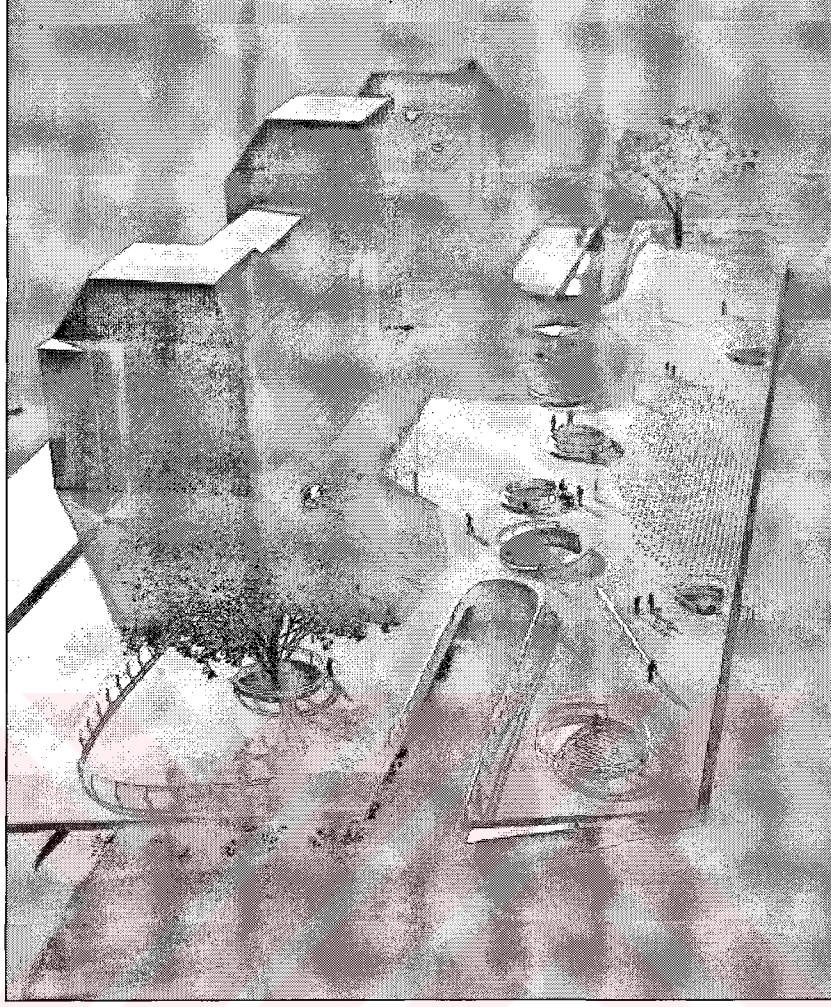


Şekil 3-13 Xp-Gen türetimlerine bir örnek (Pak, Özener, Erdem 2003)

3.1.5 Mimari Uygulamada Türetici Sistem Kullanımına Bir Örnek

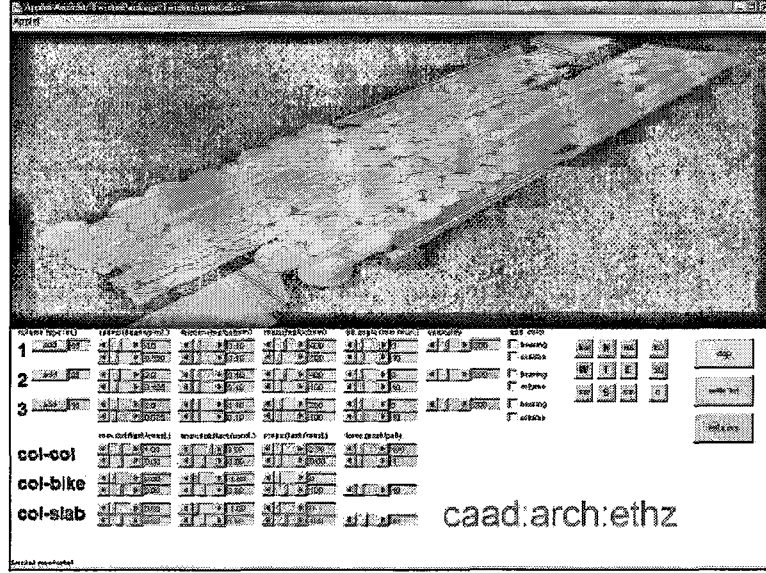
Son yıllarda türetici sistemler uygulama alanında da kullanılmaya başlamıştır. 2003 yılında Rotterdam'dan KCAP Mimarlık, Amsterdam'dan Ove Arup Mühendislik ve Zürih ETH Üniversitesi CAAD bölümü ortaklığında geliştirilen Gröningen/Hollanda Yaya Yolu projesi bu alanda uygulamaya geçmiş önemli bir örnektir. Projenin amacı Gröningen Tren İstasyonu ile şehir merkezini birbirine bağlayacak platformu taşıyacak bir kolon sistemi oluşturmaktır. Platformun farklı yapısı, alt kotta yer alan ağaçların platform yüzeyinde delikler açılmasını gerektirmesi ve platform altında 3000 bisikletlik bir park alanının oluşturulmak istenmesi, gerekli kolon sistemini türetebilecek bir sistemi zorunlu kılmıştır.

KCAP tarafından projelendirilen platformun son hali, alt kottaki ağaçların çıkabilecekleri çeşitli büyüklükte boşluklar ve iki kotu birbirine bağlayacak rampalar içermektedir. Ayrıca platformun bazı noktaları yere basarken bazı noktaları kolonlar üzerinde duracaktır.



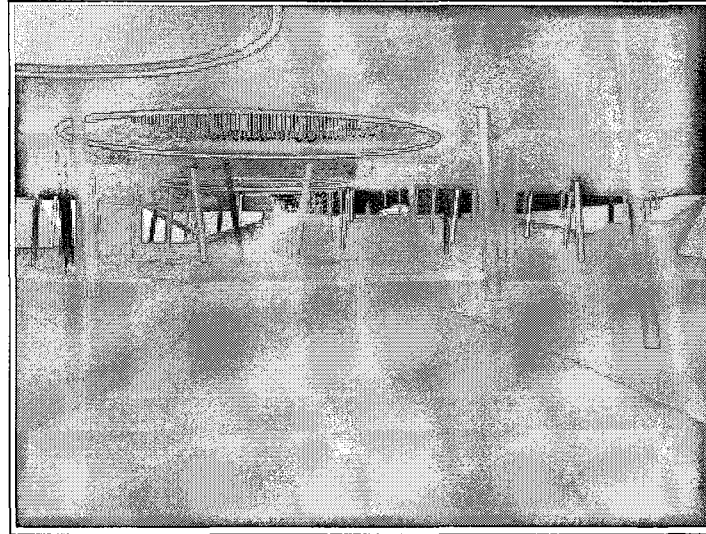
Şekil 3-14 Tasarım alternatiflerinden bir örnek maketi (Scheurer 2003)

Scheurer'in anlatımıyla (Scheurer 2003), platformun kompleks çerçevesi, önceden belirlenmiş yollar ve bisiklet park alanları, kolonların düzenli sıralanmalarını imkânsız hale getirmiştir. Bu nedenle rastgele yerleştirilmiş ve farklı açılarda eğik duran kolonlardan oluşan bir kolon ormanı fikri doğmuştur. Ancak parametrelerin ve alabilecekleri değerlerin çok fazla olması, bu işi Arup'un belirleyeceği fiziksel sınırlamalar dahilinde çözebilecek bir üretici programı gerektirmiştir. Bu program aracılığıyla Arup mühendisleri, kolonların etki alanlarını, kolonların yoğun olması gereken yerleri enteraktif olarak belirleyebilmişler, belirli bir kolonun, kalınlık, eğrilik gibi değerlerini değiştirerek tüm sistem üzerindeki etkileri gözlemleyebilmişlerdir.



Şekil 3-15 Geliştirilen programın arayüzü (Scheurer 2003)

Projeden kullanılan aracın en temel özelliği interaktivitedir (Scheurer 2003). KACP'nin tasarım kriterleriyle Arup'un yapısal kısıtlamalarını basit bir simülasyon sistemine dönüştürerek, kullanıcının işinin sadece işlemi başlatıp durdurmak olduğu bir araç yerine, kullanıcının istediği an sonuçları etkileyebileceği yüksek derecede enteraktif bir araç tercih edilmiştir. Bu tercihle üretici sistem basit bir optimizasyon aracı olmaktan çıkıp gerçek bir tasarım aracı haline gelmiştir.



Şekil 3-16 Tasarlanan platformun, alt kottan elde edilen, üç boyutlu perspektif çizimi (Scheurer 2003)

4. BİR TÜRETİCİ SİSTEM DENEMESİ: ConGen

Araştırma sırasında geliştirilen program, 3ds Max programıyla çalışan bir macroscript* uygulamasıdır. Discreet firmasına ait bu programı kullanmanın platform bağımlılığı (program yalnızca Windows işletim sisteminde kullanılabilir), pahalılık (3ds Max programının bugünkü değeri 5000\$'dır.) gibi sakıncaları bulunmaktadır. Ancak bu program tasarım çevrelerinde, modelleme ve sunum aracı olarak, yüksek bir kullanım oranına sahiptir. Ayrıca nesne yönelimli** altyapısı, buna bağlı olarak oluşturulmuş hazır geometri sınıfları***, geliştirilen aracın programa entegrasyonunu kolaylaştırmaktadır. Bu nedenlerle araştırmanın bu aşamasında 3ds Max programı tercih edilmiştir.

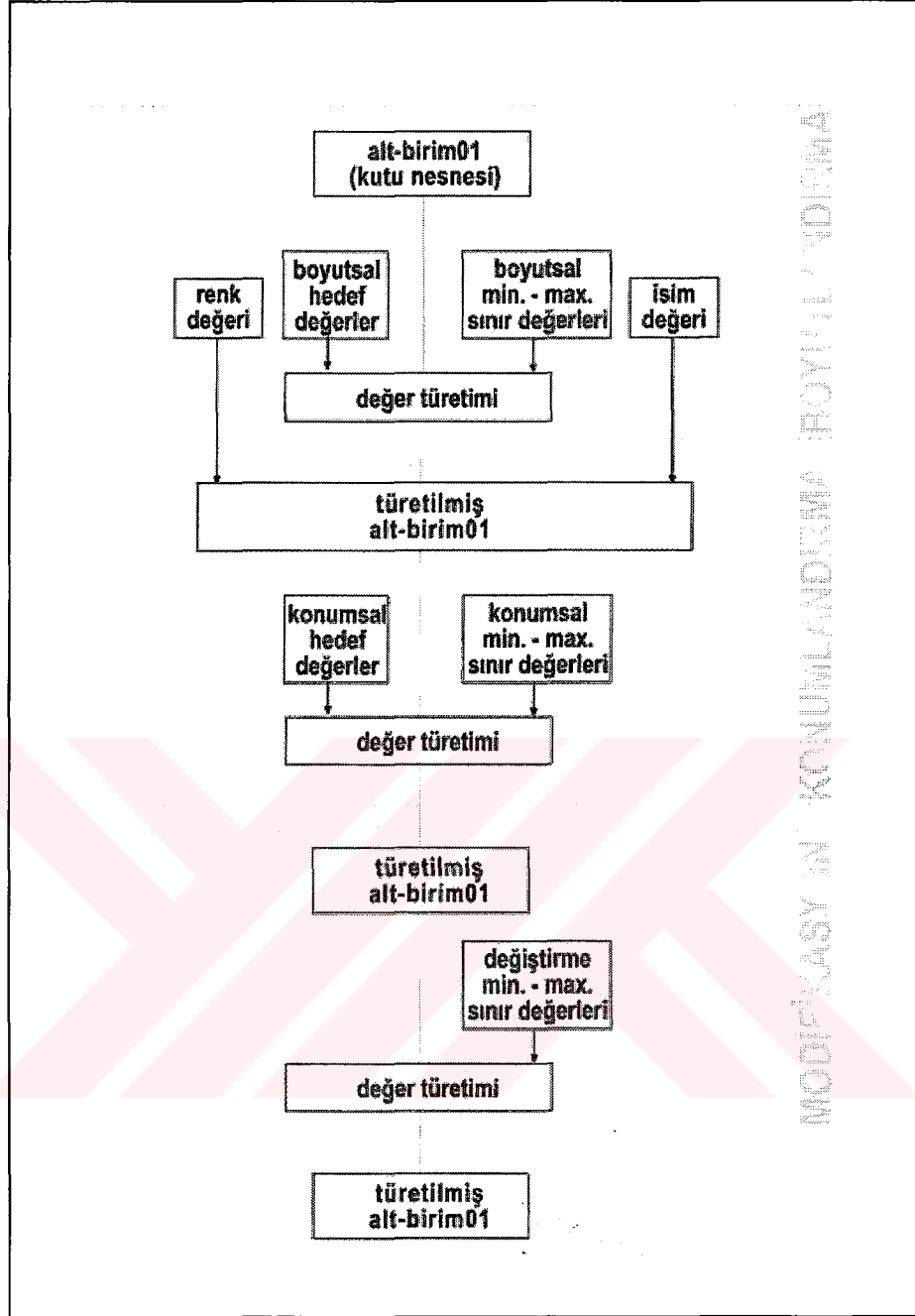
Geliştirilen programın, tasarım sürecinin erken aşamalarında, form geliştirme aşamasında kullanılması öngörüldüğü için, programa, ConGen -constraint based parametric form generator- (sınır tabanlı parametrik form üreticisi) adı verilmiştir. ConGen'de form üretim mekanizmasının temel mantığı, kullanıcı tarafından belirlenmiş değerler arasında kalacak rastgele değerlerin türetilmesidir. Bu değerler yine ConGen tarafından ana formu oluşturan alt-birimlerin (sub-objects) boyut ve pozisyon parametrelerine atanacaktır.

ConGen programında, kullanıcının tasarlayacağı nesnenin farklı pozisyonlara ve büyüklüklere sahip alt birimlerden oluştuğu varsayılır. Bu alt birimler bir evin farklı fonksiyonlara sahip bölümleri ya da bir koltuğun farklı malzemelere sahip parçaları olabilirler. Kullanıcının, alt-birimler için belirlediği, karşılıklı etkileşime dayalı, pozisyon ve boyutlara ait sınır değerleri, ConGen'in, içinde üretim yapacağı alternatif uzayını belirlerler. Bu bilgiler, program için hazırlanmış arayüz aracılığıyla alt birimlerin belirli özelliklerine atanırlar.

* 3ds Max programında çalışılırken yapılan işlemlerin, programın yazılım diline çevrilerek kaydedildiği, ya da yapılacak işlemlerin bir metin editörü ile yazılabileceği özel bir yazılım dilidir.

** Nesne yönelimli programlamada, bir yönergeler listesinden oluşan bilgisayar programının tersine birbirinden bağımsız özelliklere sahip nesnelere oluşan bir program anlayışı vardır. Bu nesnelere, belirli mesajları alabilir, belirli verileri işleyebilir, belirli mesajlar gönderebilirler. Bu sayede program çok daha esnek bir yapıya sahip olur.

*** Nesne yönelimli programlamada, bir nesne ailesinin özelliklerini, hangi mesajları alıp verebileceğini, hangi verileri işleyebileceğini belirleyen ana nesneye, o nesne ailesinin sınıfı denir.

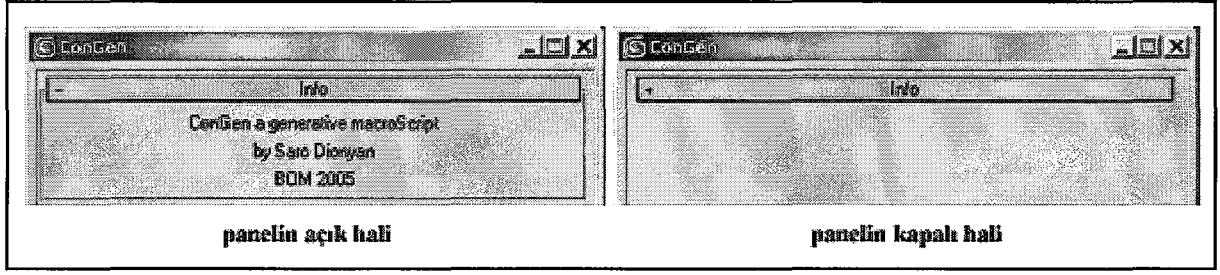


Şekil 4-1 ConGen programının akış diyagramı

4.1 ConGen Arayüzü

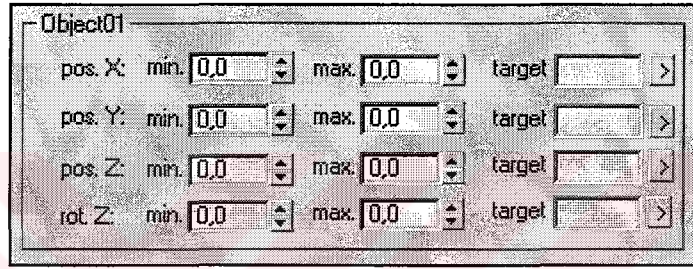
ConGen arayüzünde 3ds Max programının yerleşik arayüz sınıfları (built-in interface classes) kullanılmıştır. Programda kullanılan arayüz bileşenleri sekiz tiptir:

Katlanabilir Panel (Rollout-panel) : Diğer arayüz bileşenlerini üzerinde taşıyan, istenildiğinde kapatılarak sadece başlığı görünebilen, ana arayüz elemanıdır.



Şekil 4-2 Katlanabilir panelin açık ve kapalı halleri

Parametre Grupları (Group-Box): Çeşitli arayüz bileşenlerini tek bir başlık altında gruplamak amacıyla kullanılır.



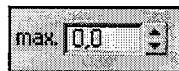
Şekil 4-3 Object01 adıyla etiketlenmiş bir parametre grubu.

Etiket (Label): Kullanıcıya bilgi vermek amacıyla kullanılan, kullanıcının değiştirme seçeneğinin olmadığı, metin parçalarıdır.



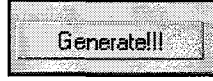
Şekil 4-4 Bir etiket örneği (pos. X).

Çevirici (Spinner): Program kodu içerisinde belirlenmiş sınırlar arasında, istenilen değerin klavye ya da çevirici okları aracılığıyla belirlenmesini sağlar



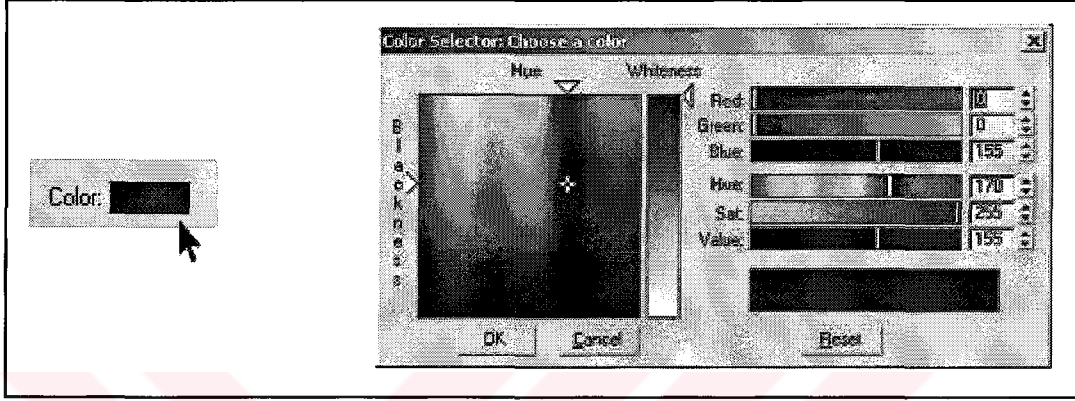
Şekil 4-5 Bir çevirici örneği.

Düğme (Button): Kendisiyle ilişkilendirilmiş işlemin aktive olmasını sağlar.



Şekil 4-6 Bir düğme örneği.

Renk Seçim Düğmesi (colorPicker) : Renk seçimi için kullanılır. Tıklandığında, bir Windows bileşeni olan “Color Selector” aracı açılır.



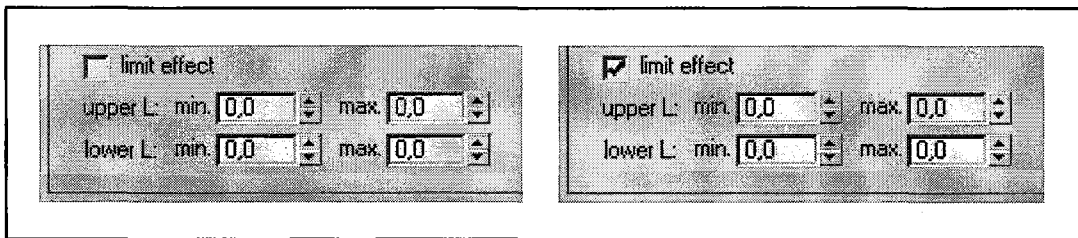
Şekil 4-7 Renk seçim düğmesi ve renk seçimi arayüzü

Değiştirilebilir Metin Kutucuğu (Edit-Text): Kullanıcının klavye aracılığıyla metin yazabilmesi için kullanılır.



Şekil 4-8 Değiştirilebilir metin kutucuğu örneği.

Onay Kutucuğu (Check Box): Kendisine bağlı bir değişkeni doğru ya da yanlış hale getirir.

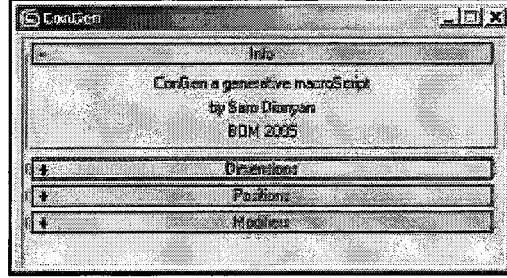


Şekil 4-9 Onay kutucuğunun işaretlenmemiş ve işaretlenmiş durumu (limit effect)

Bu sekiz arayüz bileşenin farklı şekillerde kullanımıyla oluşturulan ConGen arayüzü, dört katlanabilir panelden oluşmaktadır:

1.- İno Paneli:

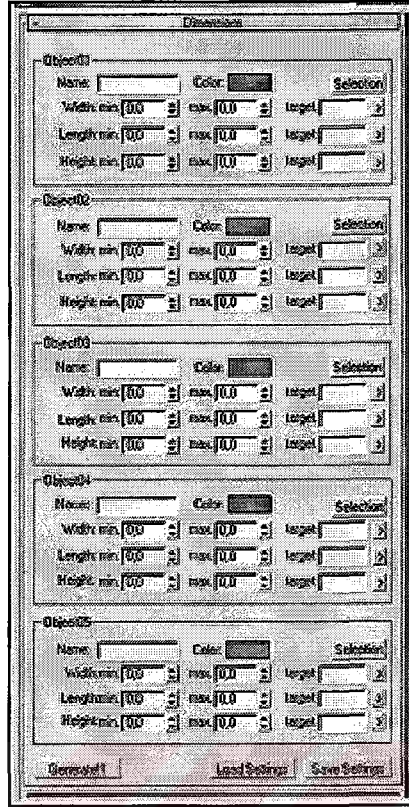
Bu panel, programı yazanın kimliđi, yapım yılı ve versiyonu ile ilgili bilgiler içerir.



Şekil 4-10 İno panelinin görünümü

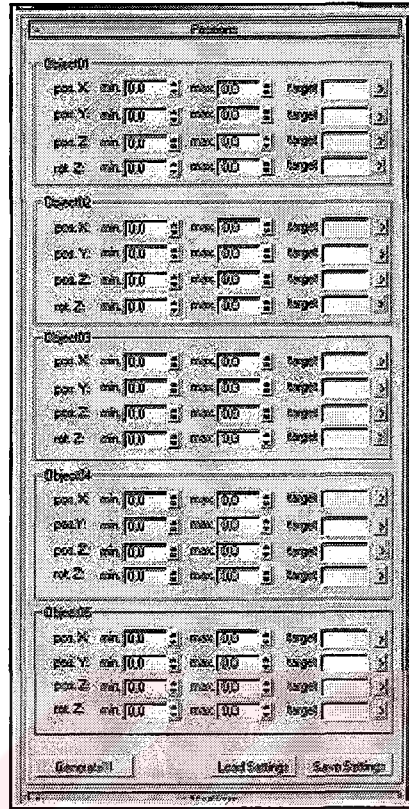
2.- Boyutlandırma (Dimensions) Paneli:

Bu panel sekiz bileşenden oluşmaktadır: a) Her alt-birimin isimlendirilebilmesi için deđiştirilebilir metin kutucuđu b) Her alt-birim için bir renk seçim düğmesi c) Her alt – birimin genişlik, uzunluk ve yükseklik parametresinin alabileceđi minimum ve maksimum deđerlerin belirlenmesi için bir çevirici. d) Alt-birimlerin, her parametresine, harici bir “script” atanabilmesi için “target” etiketli, deđiştirilebilir metin kutucuđu ve düğmesi. e) Her alt-birim için, Max programında çizilmiş herhangi bir formu seçilen alt-birime eklemleyebilmeye yarayan “Selection” (Seçim) düğmesi. f) Boyutlandırma ayarlarını kaydedebilmek için “Save Settings” düğmesi. g) Kaydedilmiş boyutlandırma ayarlarını geri yükleyebilmek için “Load Settings” düğmesi. h) Belirlenen sınırlamalar içerisinde kalacak alt-birim boyutlarının türetilmesi için “Generate” düğmesi.



Şekil 4-11 Boyutlandırma panelinin görünümü.

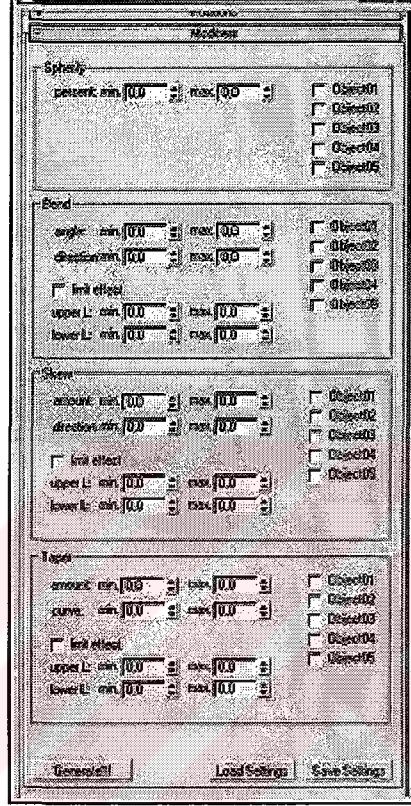
3.- Konumlandırma (Positions) Paneli: Bu panel 6 bileşenden oluşmaktadır. a) Alt-birimlerin x,y,z pozisyon parametrelerinin alabilecekleri maksimum ve minimum değerlerin belirlenmesi için çeviriciler. b) Her alt-birimün öklidyen transformasyonunun maksimum ve minimum derecesinin belirlenebilmesi için çeviriciler. c) Alt-birimlerin, her parametresine, harici bir “script” atanabilmesi için “target” metin kutucuğu ve düğmesi. f) Pozisyon ayarlarını kaydedebilmek için “Save Settings” düğmesi. g) Kaydedilmiş pozisyon ayarlarını geri yükleyebilmek için “Load Settings” düğmesi. h) Belirlenen sınırlamalar içerisinde kalacak alt-birim pozisyonlarının türetilmesi için “Generate” düğmesi.



Şekil 4-12 Konumlandırma panelinin görünümü.

4- Değiştiriciler (Modifiers) Paneli: ConGen’de kullanılmak için örnek olarak seçilmiş 3ds Max araçları olan değiştiricilere ait ayarlar bu panelde yer alırlar. Panelde dört ana grup bulunur: a) Küreselleştirme (Spherify) grubu iki bileşenden oluşur. Bunlar, minimum ve maksimum küreselleştirme oranını belirleyen çeviriciler ve bu değiştirmenin hangi alt-birimlere uygulanacağını belirleyen onay kutucuklarıdır. b) Bükme (Bend) grubu dört bileşenden oluşur. Bunlar, bükülme açısı ve yönüne atanabilecek minimum ve maksimum değerleri belirleyen çeviriciler, değiştirici limitlerinin aktive olmasını sağlayan “limit-effect” etiketli onay kutucuğu, değiştirici limitlerinin alabilecekleri maksimum ve minimum değerleri belirleyen çeviriciler ve değiştiricinin hangi alt-birimlere uygulanacağını belirleyen onay kutucuklarıdır. c) Çarpıtma (Skew) grubu dört bileşenden oluşur. Bunlar, çarpılma miktarına ve yönüne atanabilecek minimum ve maksimum değerleri belirleyen çeviriciler, değiştirici limitlerinin aktive olmasını sağlayan “limit-effect” etiketli onay kutucuğu, değiştirici limitlerinin alabilecekleri maksimum ve minimum değerleri belirleyen çeviriciler ve değiştiricinin hangi alt-birimlere uygulanacağını belirleyen onay kutucuklarıdır. d) Konikleştirme (Taper) grubu dört bileşenden oluşur. Bunlar, koniklik miktarına ve eğrilğine atanabilecek minimum ve maksimum değerleri belirleyen çeviriciler, değiştirici limitlerinin

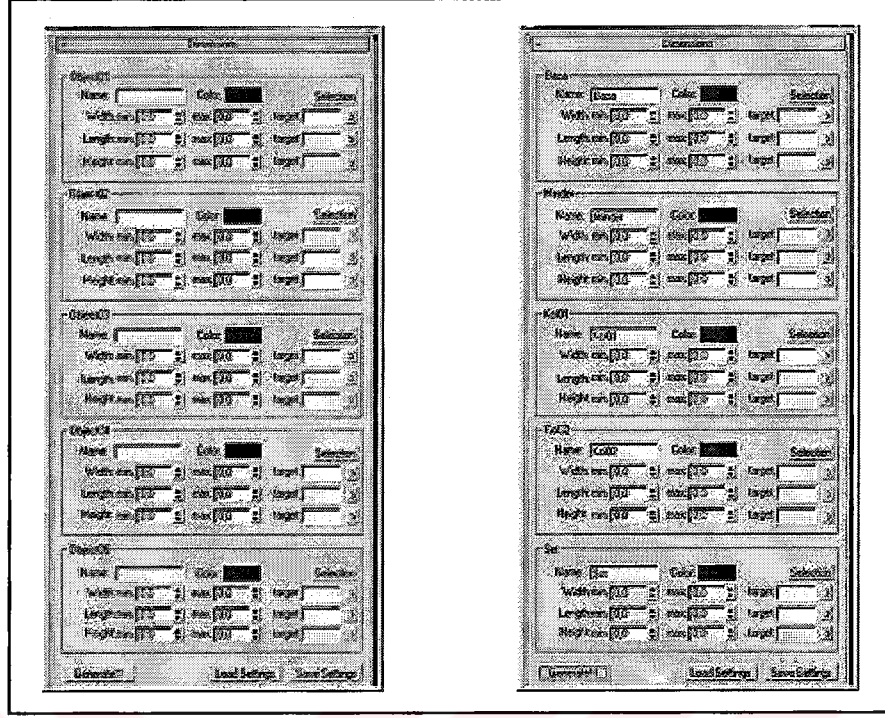
aktive olmasını sağlayan “limit-effect” etiketli onay kutucuğu, değıştirci limitlerinin alabilecekleri maksimum ve minimum değeri belirleyen çeviriciler ve değıştiricinin hangi alt-birimlere uygulanacağını belirleyen onay kutucuklarıdır.



Şekil 4-13 Değıştiriciler panelinin görünümü.

4.2 Alt-birimlerin İsimlendirilmesi

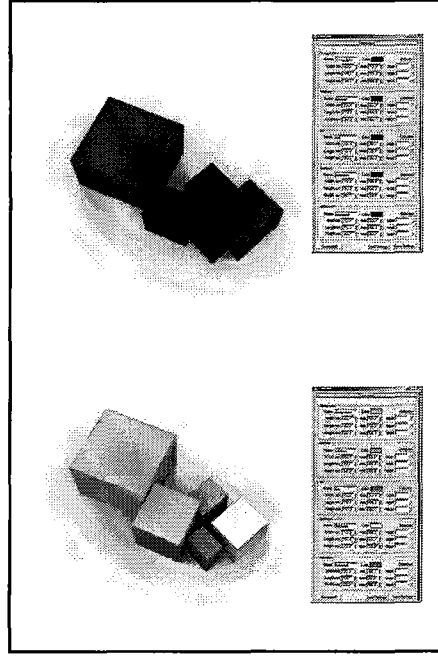
ConGen programı çalıştırıldığında, program kodunda varsayılan (default) ayarlara dayanarak üretilen birimler “Object01...Object05” şeklinde adlandırılırlar. Ancak kullanıcı tasarlanacak nesnenin alt-birimlerini boyutlandırma panelinde yer alan, “Name:” etiketine sahip metin kutucuğunu kullanarak yeniden isimlendirebilir. İstenilen isimler girildikten sonra her paneldeki “Generate” düğmesine basıldığında, isim etiketleri güncelleneceklerdir.



Şekil 4-14 Alt-birimler isimlendirilmeden önce ve sonra arayüz görünümüleri

4.3 Alt-birimlerin Renklendirilmesi

Program çalıştırıldığında, üretilen alt-birimler program kodunda varsayılan mavi renge sahiptirler. Kullanıcı, alt-birimlere farklı renkler vermek istediğinde, boyutlandırma panelinde bulunan renk seçim düğmelerine tıkladığında açılan “Color Selector” aracının yardımıyla istenilen rengi seçebilir. Renk seçimi yapıldığında, renk seçim düğmesinin rengi seçilen renge dönüşecektir. Bu işlem tamamlandıktan sonra “Generate” düğmesine basılarak sahnedeki alt-birimler istenilen şekilde renklendirilmiş olacaktır.



Şekil 4-15 Alt-birimlerin renklendirilmesi ve arayüzün güncellenmesi

4.4 Alt-birim Boyutlarının Ayarlanması

ConGen programında, alt-birimler 3dMax programında hazır olarak bulunan Küp sınıfına aittirler. Bu sınıfın 7 temel özelliği vardır: 1- Genişlik 2-Uzunluk 3-Yükseklik 4-Malzeme 5-Genişlik Segment Adedi 6-Uzunluk Segment Adedi 7-Yükseklik Segment Adedi.

Segment adetleri, küp yüzeylerinin kaç yüzey parçasından (dolayısı ile kaç köşe ve kaç kontrol noktasından) oluşacağını belirler. Bu özellik daha sonra o yüzeylerin geliştirilmesiyle ilgili olduğu ve araştırmanın kapsamına girmediği için, segment adedine ait özellikler program arayüzüne alınmamıştır.

Boyutlandırma panelinde, kullanıcının sınır değerlerini belirleyebileceği üç alt-birim özelliği bulunur. Bunlar genişlik, uzunluk ve yüksekliktir. Malzeme özelliği ise yalnız renk seçimi ile sınırlandırılmıştır. Genişlik, uzunluk ve yükseklik (width, length, height) parametrelerinin “min” ve “max” çeviricilerine farklı değerler girilip, “Generate” düğmesine basıldığında, program bu iki değer arasında bulunan rastgele bir değer üretecek ve ilgili parametreye atayacaktır.

Object01.width = random o01_w_min. value o01_w_max. value

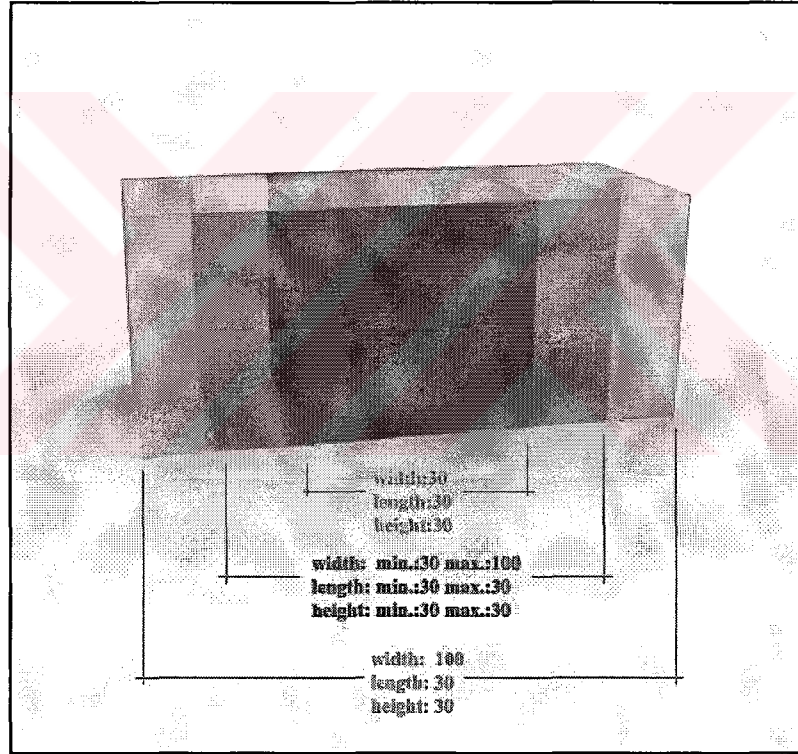
Yukarıdaki kod, “object01” alt-biriminin genişliği için, arayüzde belirlenmiş “min.” ve “max” değerleri arasında rastgele bir değer üretecektir. Aşağıdaki tabloda “min.” için 30 “max.” için

100 değeri girildiğinde genişlik parametresine atanabilecek değerler görülmektedir.

Çizelge 4-1 Farklı min ve max değerleri için üretilmiş değer örnekleri

Alt-birim genişliği için Min. Ve Max. Değeri	Türetilen alt-birim genişliği
30 - 100	51
40 - 40	40
10 - 40	20
40 - 10	26

Herhangi bir parametrenin, “min” ve “max” çeviricilerine aynı değer yazıldığında, rastgele seçim aralığı oluşmadığı için o parametreye “min” ve “max” çeviricilerine girilen değer atanacaktır.



Şekil 4-16 Genişlik parametresi için minimum 30 maksimum 100 değeri girilen nesnenin, minimum, maksimum ve türetilebilecek değerler aralığının gösterimi

4.4.1 Hedeflenen Boyut Değerlerinin Türetimi İçin Programa Kod Eklenmesi

Boyutlandırma panelinde alt-birimlere ait her parametre ayarının sonunda “Target” etiketli bir metin kutucuğu ve bu kutuya bağlı ok işaretli bir düğme bulunur (bu düğmeye basıldığında metin yazımı ve okunuşunu kolaylaştırmak için eklenmiş bir metin penceresi açılacaktır). Bu metin kutucuğuna max-script dilinde bir kod yazıldığında, bu kod çalıştırılarak üretilen değer ilgili parametreye atanacaktır. Bu özellik alt-birimler arasında mantıksal ilişkiler kurabilmek

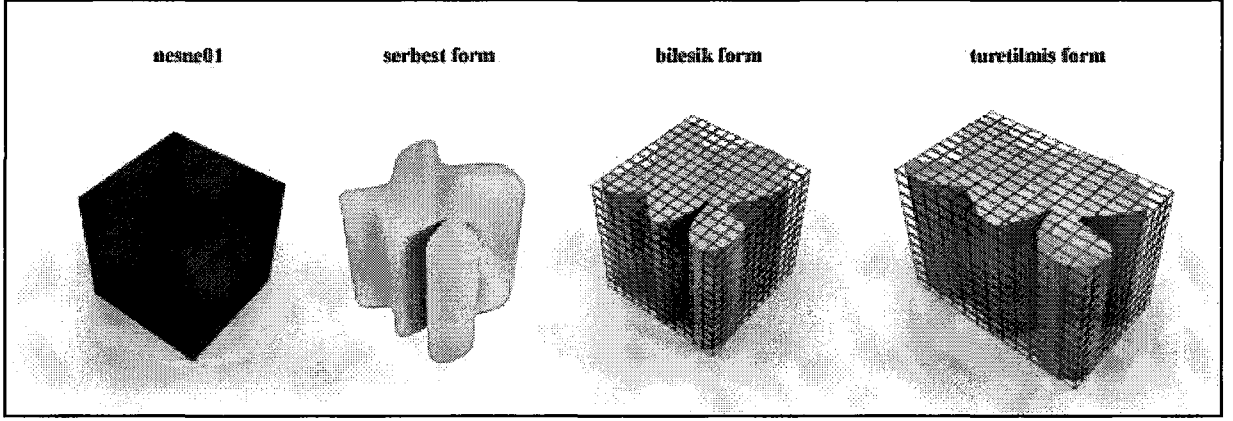
amacıyla eklenmiştir. Ayrıca bu özellik sayesinde ConGen farklı tasarım problemleri için kullanılabilme olanağına sahip olmuştur. Aşağıdaki tabloda programa eklenebilecek kodlar ve bunların işlevlerine örnekler verilmiştir.

Çizelge 4-2 ConGen'e eklenebilecek kod örnekleri ve bunların işlevlerine örnekler

	Min.	Max.	Target	Kod Anlamı
Object01.width	30	100	-	
Object02.width			if \$Object01.width>60 \$Object02.width=50 else \$Object02.width=\$Object01.width+20	Object01 genişliği 60'dan fazla ise object02 genişliği 50, değil ise object01 genişliğinden 20 fazla olacaktır
Object03.width	100	150	-	-
Object04.width			500-\$Object01.width- \$Object03.width	Object04 genişliği object01 ve object03'ün 500den çıkarılmasıyla belirlenecektir
Object05.width			(\$extern01.max.x-\$extern01.min.x)- 50	Program dışında çizilen extern01 adlı şeklin genişliğinden 50 az bir değer object05'in genişliğine atanacaktır.

4.4.2 Max Programında Üretilmiş Şekillerin ConGen Nesneleriyle İlişkilendirilmesi

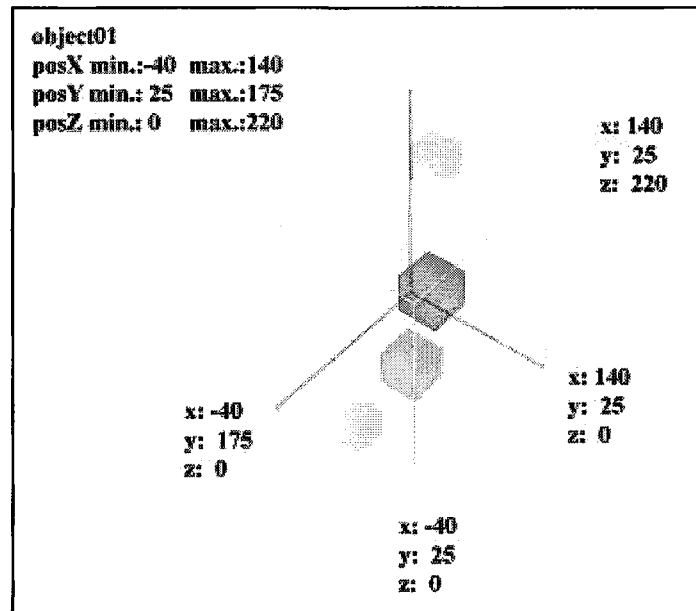
ConGen programı form türetiminde küplerden faydalanmaktadır. Ancak kullanıcı, farklı formlar kullanmak istediğinde, 3ds Max programının araçlarıyla geliştirilmiş herhangi bir şekli, ConGen'in alt-birimleriyle birleştirebilir. Bu işlem, boyutlandırma panelinde yer alan "Selection" düğmeleriyle gerçekleştirilir. Birleştirmek istenen şekil seçiliyken, belirlenen alt-birimin parametre grubunda yer alan "Selection" düğmesine basıldığında, seçili şekil o alt-birimle birleşecektir. Bu işlemden sonra, birleştirilen iki şekil, bir ağ (mesh) nesnesi haline geleceklerdir. Bu nesnenin genişlik, uzunluk, yükseklik gibi parametreleri yoktur, bu parametrelerle ilgili daha önce belirlenmiş kısıtlamalar, program kodu içerisinde x,y,z ölçeklendirmesiyle sağlanırlar. Pozisyon parametreleri ise bileşik şeklin parametrelerine atanabileceklerdir.



Şekil 4-17 ConGen'de serbest formların kullanılması

4.5 Alt-Birim Pozisyonlarının Ayarlanması

ConGen arayüzünde alt-birim pozisyonlarının ayarlanmasına ilişkin bileşenler konumlandırma panelinde bulunurlar. Bu panelde kullanıcının sınır değerlerini belirleyebileceği dört parametre bulunur. İlk üç parametre, alt-birimlerin x-y-z koordinat sistemindeki konumlarına ilişkindir. Dördüncü parametre ise alt-bimirlere uygulanacak öklidyen transformasyon miktarını belirler. 3ds Max programında, üretilen herhangi bir formun varsayılan pozisyonu ağırlık merkezinin yerine göre belirlenir. Kullanıcı tarafından belirlenen sınır değerler, o nesne merkezinin içinde hareket edebileceği sanal bir çerçevenin boyutlarıdır.



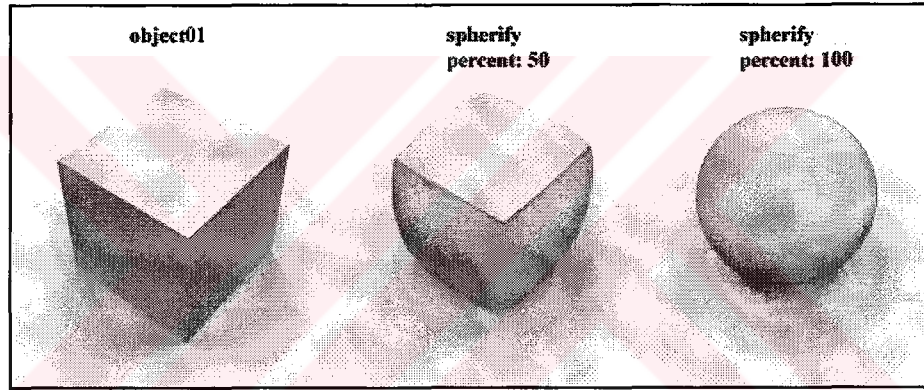
Şekil 4-18 Pozisyon sınırları için girilmiş değerlerin oluşturduğu sanal çerçeve.

4.6 Deęiřtiricilerin Uygulanması

ConGen arayüzünde, örnek olarak alınmış dört 3ds Max deęiřtiricisinin seçilmiş parametreleri için sınır deęerler girme olanaęı sağlanmıştır. Bu deęiřtiricilerin etkisine program kodu içinde herhangi bir müdahalede bulunulmamıştır. Dört deęiřtiricinin de uygulama şekilleri ve buna ilişkin parametreleri birbirinden farklıdır:

1.- Küreselleřtirme

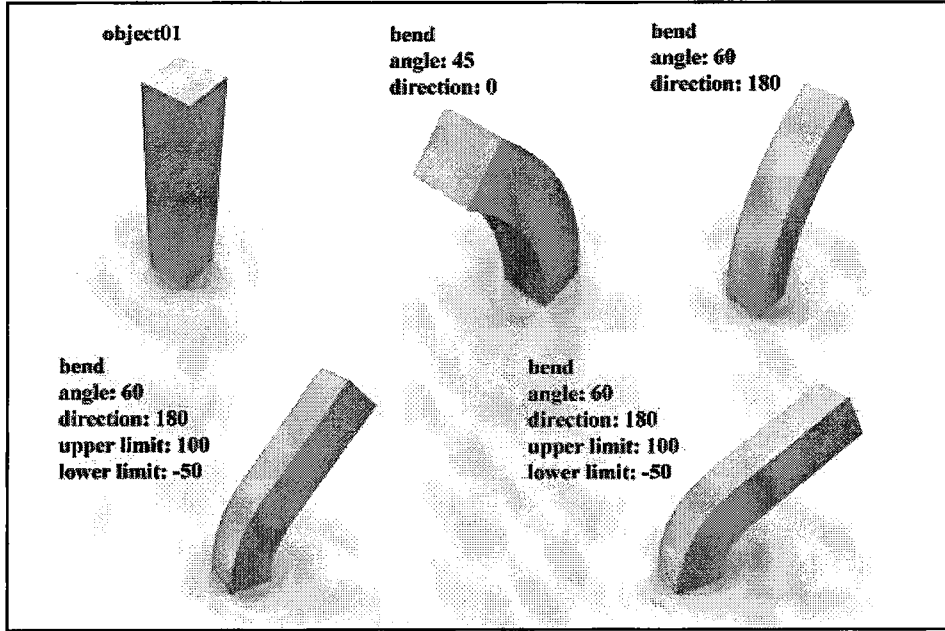
Bu deęiřtirici, uygulandıęı nesneyi, boyutlarının izin verdięi ölçüde küre haline dönüřtürür. Küreselleřtirme oranını belirlemeye yarayan, tek bir parametresi vardır. ConGen'de kullanıcı küreselleřtirme oranı için minimum ve maksimum deęerleri belirleyip, programın bu deęerler arasındaki alternatifleri türetmesini sağlar.



Şekil 4-19 Bir küpe farklı oranlarda küreselleřtirme deęiřtiricisinin uygulanmasına örnekler.

2.- Bükme

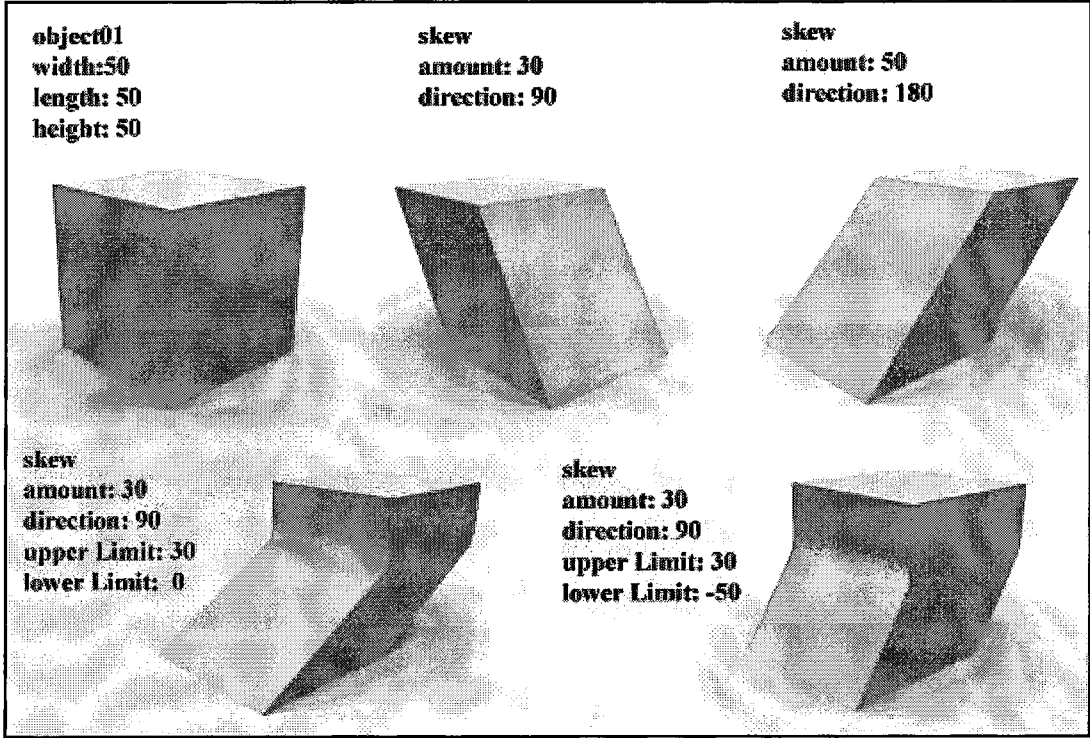
Bu deęiřtirici, uygulandıęı nesneyi belirlenen miktarda ve yönde bükür. Nesnenin, hangi açıda büküleceęini belirleyen açı (angle) parametresi, hangi yönde büküleceęini belirleyen yön (direction) parametresi ve bükülmenin nesnenin hangi bölümüne uygulanacaęının belirleyen alt ve üst limit parametreleri ConGen arayüzüne alınmıştır. Bükülmenin hangi aksta yapılacaęını belirleyen X-Y-Z onay kutucukları ConGen arayüzüne alınmamıştır. Program kodu içerisinde bükülme aksı Z olarak belirlenmiştir.



Şekil 4-20 Bir prizmaya uygulanan bükme değıştiricisinin etkilerine örnekler.

3.- Çarpıtma

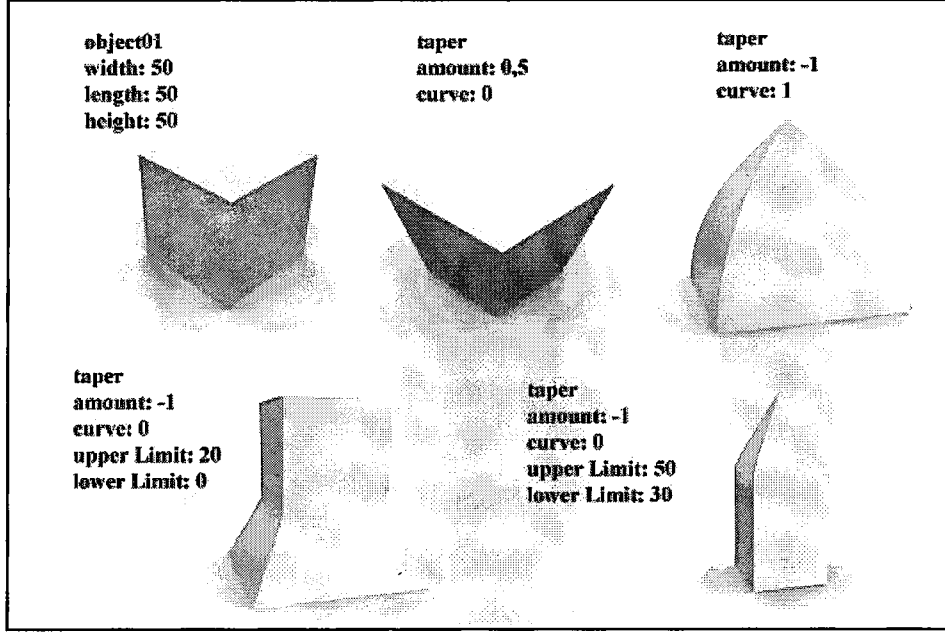
Bu değıştirici, uygulandıđı nesneyi belirlenen miktarda ve yönde çarpıtır. Nesnenin, ne miktarda çarpıtılacağını belirleyen miktar (amount) parametresi, hangi yönde çarpıtılacağını belirleyen yön (direction) parametresi ve çarpıtmanın nesnenin hangi bölümüne uygulanacağını belirleyen alt ve üst limit parametreleri, ConGen arayüzüne alınmıştır. Çarpıtmanın hangi aksta yapılacağını belirleyen X-Y-Z onay kutucukları ConGen arayüzüne alınmamıştır. Program kodu içerisinde çarpıtma aksı Z olarak belirlenmiştir.



Şekil 4-21 Bir prizmaya uygulanan skew (çarpıtma) değıştiricisinin parametrelerinin etkilerine örnekler.

4.- Konikleřtirme

Bu değıştirici, uygulandıđı nesneyi belirlenen miktarda ve eğrilikte konikleřtirir. Nesnenin, ne miktarda konikleřtirileceđini belirleyen miktar (amount) parametresi, eğriliđini belirleyen eğrilik (curve) parametresi ve konikleřtirmenin nesnenin hangi bölüme uygulanacađını belirleyen alt ve üst limit parametreleri, ConGen arayüzüne alınmıřtır. İşlemin hangi aksta yapılacađını belirleyen X-Y-Z onay kutucukları ConGen arayüzüne alınmamıřtır. Program kodu içerisinde konikleřtirme aksı Z olarak belirlenmiřtir.



Şekil 4-22 Bir prizmaya uygulanan konikleştirme değıştiricisinin parametrelerinin etkilerine örnekler.

4.7 ConGen Programının Kullanım Kısıtlamaları

ConGen'in geliştirilmesi sırasında yapılan tercihler ya da 3ds Max script dilinin gereklilikleri nedeniyle programın kullanımından çeşitli kısıtlamalar bulunmaktadır:

1. Program açılışında otomatik olarak üretilen beş alt-birim (kutu nesnesi) program kapatıldığında otomatik olarak sahneden silinmeyecektir. Bu işlem gerektiğinde kullanıcı tarafından gerçekleştirilmelidir.
2. Alt-birimlere ait değerlerin kullanıcı tarafından kesin olarak belirlenmesi istendiğinde arayüzde bunu sağlayacak özel bir bölüm bulunmamaktadır. Kullanıcı istenilen değeri ait olduğu parametrenin min. ve max. limitlerine yazmalıdır.
3. Hedef kodlar (Target Scripts) aracılığıyla alt-birimler arasında ilişki kurulmak istendiğinde bu ilişki hiyerarşik olarak kurulmalıdır. Örneğin, Nesne01 genişliği için Nesne03 genişliği hedef gösterilemez ancak tersi geçerli olabilecektir.
4. Her panel için ayrı ayrı bulunan kaydetme ve yükleme seçenekleri yalnızca o paneldeki parametrelere ait değerler ve hedef kodlar için çalışmaktadırlar. Sahne üzerinde yapılacak değışiklikler ve birleştirmeler kayıt edilemeyeceklerdir.

5. ConGen için seçilmiş olan küreselleştirme dışındaki 3ds Max değiştiricileri yalnızca z aksında çalışabilecek şekilde program içine alınmışlardır.
6. Alt-birimlere uygulanan herhangi bir değiştiricinin etkisi kaldırılmak istendiğinde, o değiştiricinin istenilen parametresine sıfır değeri verilip türetim yapılmalıdır. Etkiyi kaldırabilecek ayrı bir seçenek program arayüzünde bulunmamaktadır.

4.8 ConGen Programı ile Tasarım Türetmek için Önerilen Çalışma Metodu

ConGen programının desteği ile tasarım için önerilen metot, temel olarak genel kararları verilmiş bir parametrik formun istenen sınırlar içerisinde alternatiflerinin türetilip değerlendirilmesine dayanır. Bu işlem altı adımda ele alınabilir:

1.- Varolan Tasarım Çözümlerinin Değerlendirilmesi

Parametrik form türetimi rastgele sayı türetimine ya da eklenen kodların hedeflediği değerlere dayanır. Bu noktada kullanılan temel bilgi kullanıcı tarafından, belirli parametreler için yazılan sınır değerler ya da kodlardır. Bu noktada kullanılan bilginin kaynağı, çalışılmak istenen alanda daha önce üretilmiş çözümlerdir. Tasarımcı geliştirmek, alternatifler üretmek istediği çözümlerden bir set oluşturup, buradan elde ettiği sınır değerleri kendi hedefleri doğrultusunda değerlendirmelidir.

2.- Limitlerin Ve Karşılıklı İlişkilerin Oluşturulması:

Alt-birimlerin birbirleriyle pozisyonel ve boyutsal ilişkisi, genel bir form kabulü çerçevesinde kurulur. Belirlenen sınır değerler ya da ek kodların etkileri türetim yapılarak gözlenir. İstenilen alternatifleri oluşturabilmek amacıyla sınır değerlerde ya da hedef kodlarda değişiklikler yapılarak yeniden türetim aşamasına geçilir. Bu süreç istenilen değer aralıklarına ulaşıncaya kadar döngüsel olarak tekrarlanır.

4.- Form Türetimi:

ConGen'de alt-birimlerin boyutları için yapılan türetim ile pozisyonlar için yapılan alternatif türetimi birbirinden ayrı olarak ele alınmıştır. Bu sayede alt-birimlerin boyutlarına ait sınır değerlerin etkisi ve konumlandırmaya ait sınır değerlerin etkisi ayrı ayrı ele alınarak gözlemlenebilir. Bu aşamada program, sınır değerler arasında kalacak rastgele değerler üretecek, ya da hedef kodların sonuçlarını parametrelere atayacaktır.

5.- Form Seçimi:

Form seçimi, kullanıcı ihtiyaçlarına, malzeme kriterlerine ya da uzman görüşlerine dayanılarak yapılabilir. 3d Max programının özellikleri kullanılarak, gerçekçi çizimler üretilip türetilen formların değerlendirilmesi kolaylaştırılabilir. Yine 3ds Max programının animasyon araçlarından faydalanılarak, üretim alternatifleri arasındaki farkların oluşumu ve etkisi detaylı olarak gözlemlenebilir.

6.- Tasarımın Detaylandırılması:

Birden beşe kadar olana adımlar birbirinden farklı örnekler geliştirilerek tekrarlanabilir. Elde edilen sonuçlara yeni parametrelerin eklenmesi, bazı birimlerin detaylandırılması ile tasarım sonuçlandırılır.



5. CONGEN İLE ÜRETİLMİŞ TASARIM ÖRNEKLERİ

Bu bölümde, geliştirilen ConGen programı ile türetilmiş tasarım örnekleri anlatılmaktadır. ConGen tasarlanırken, form geliştirmeye ilgili, birbirinden farklı tasarım dallarında kullanılabilmesi amaçlanmıştır. Bu nedenle verilecek örnekler, farklı tasarım dallarında ConGen kullanımının ne gibi sonuçlar doğurabileceğini gösterebilmek amacıyla seçilmişlerdir. Örneklerin sunumunda seçilen genel yöntem şu şekildedir:

- 1- Tasarım nesnesinin programın çalışma mantığına nasıl uyarlandığının anlatımı
- 2- Tasarım sırasında kullanılan parametre değerlerinin tablo şeklinde gösterimi
- 3- ConGen'in türettiği alternatif örneklerinin gösterimi
- 4- Türetilen örneklerin tartışılması

Farklı tasarım alanlarından seçilen örneklerin gösteriminden sonra. Bir örnek detaylı olarak ele alınacak, önerilen çalışma metodolojisini izleyerek yapılan çalışma anlatılacaktır.

5.1 Ürün Tasarımı Örneği: Parfüm Kutusu

Çalışma için, parfüm kutusunun üç alt birimden oluştuğu varsayıldı: a) Tüp (tube) b) Tüp Ağızı (tubeM) c) Kapak (lid). Bu üç alt-birimin birbirleriyle pozisyonel ve boyutsal ilişkileri için aşağıdaki değerler verildi. (Tüm değerler "cm." cinsinden düşünülmüştür.)

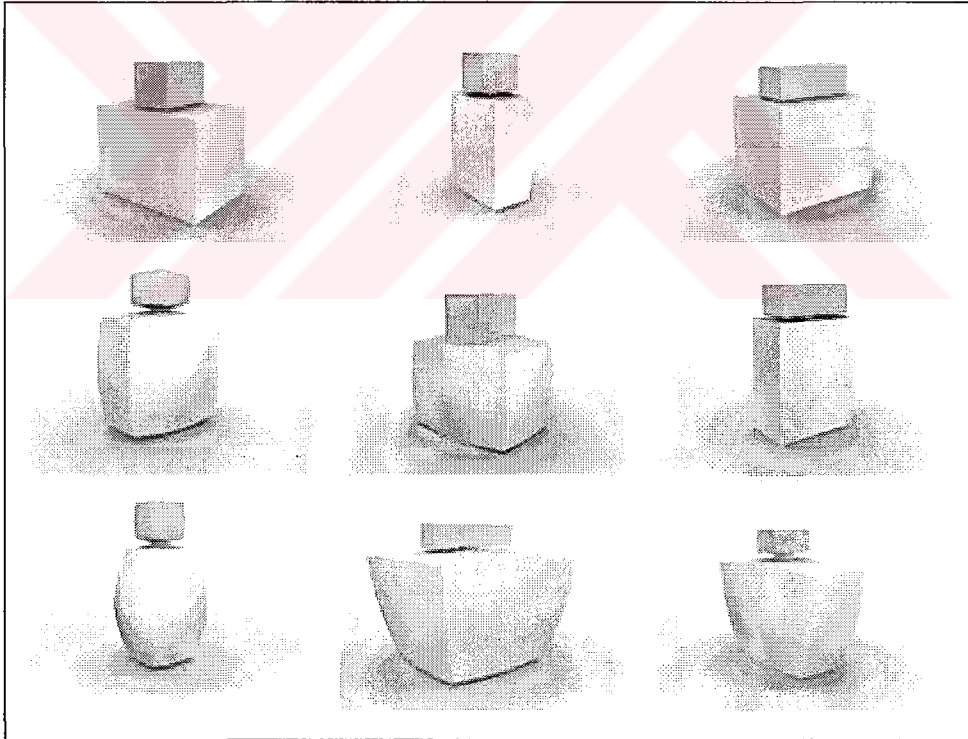
Çizelge 5-1 Parfüm kutusu alt-birim parametrelerine atanan değerler

Alt-birim parametreleri	Min.	Max.	Target	Kod Anlamı
tube. width	5	15	-	
tube.length	5	10		
tube.height	10	15		
tube.pos.x	-	-		Varsayılan değer 0'dır
tube.pos.y	-	-		Varsayılan değer 0'dır
tube.pos.z	-	-		Varsayılan değer 0'dır
tubeM.width	1	2		
tubeM.length	1	2		
tubeM.height	1	3		
tubeM.pos.x	-	-		
tubeM.pos.y	-	-		
tubeM.pos.z	-	-	\$tube.height	Tüpün yükseklik değeri atanacaktır.
lid.width	2	5		
lid.length			random (\$tube.width-2) 5	Kapağın uzunluğuna, tüpün genişliğinden 2cm

				azdır değerle 5 cm arasında bir değer atanacaktır.
lid.height	2	5		
lid.pos.x	-	-		Varsayılan değer 0'dır
lid.pos.y	-	-		Varsayılan değer 0'dır
lid.pos.z			Random (\$tube.max.z) (\$tube.max.z + 2)	Kapağın z pozisyonu tüpün en yüksek noktasıyla bu noktadan iki cm yüksek bir nokta arasında rastgele seçilecektir.

Çizelge 5-2 Uygulanan 3ds Max değiştiricileri için verilen sınır değerler

Modifier Parametreleri	Min.	Max.	Uygulandığı Birimler
spherify.amount	0	50	tube, tubeM
Taper.angle	0	0,5	tube



Şekil 5-1 ConGen'in türettiği alternatif tasarımlardan örnekler

Şekil 5-1'deki alternatif üretimler incelendiğinde, alt-birimlerin bağımsız olmalarının ve parametrelere verilen geniş değer aralıklarının, birbirlerinden oldukça farklı formların ortaya çıkmasını sağladığı görülür.

5.2 Ürün Tasarımı Örneği: Abajur

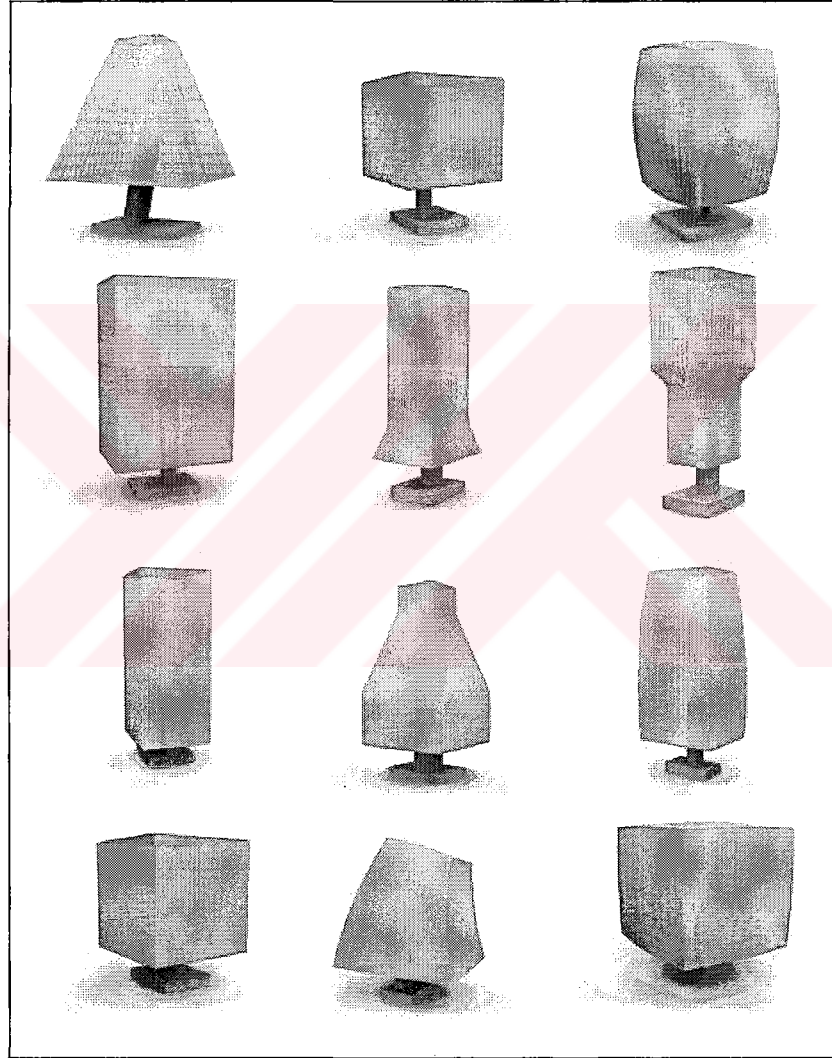
Çalışma için, abajurun dört alt birimden oluştuğu varsayıldı: a) Başlık (head) b) Ampul (bulb) c) Gövde (body) d) Taban (base) Bu dört alt-birimin birbirleriyle pozisyonel ve boyutsal parametreleri için aşağıdaki değerler verildi. (Tüm değerler “cm” cinsinden düşünülmüştür.)

Çizelge 5-3 Abajur alt-birim parametrelerine atanan değer ve ek kodlar

Alt-birim parametreleri	Min.	Max.	Target	Kod Anlamı
head.width	30	50	-	
head.length	30	50	-	
head.height	40	90	-	
head.pos.x	-	-	-	Varsayılan değer 0'dır
head.pos.y	-	-	-	Varsayılan değer 0'dır
head.pos.z	-	-	random (\$body.height+\$base.height-5) (\$body.height+\$base.height-20)	Gövde yüksekliği ile taban yükseklik değerinin toplamının 5 eksi değeri ile 20 eksi değeri arasında rastgele seçilecek
bulb.width	5	5	-	
bulb.length	5	5	-	
bulb.height	5	5	-	
bulb.pos.x	-	-	\$body.pos.x	Gövdenin X pozisyonu atanacak
bulb.pos.y	-	-	\$body.pos.y	Gövdenin Y pozisyonu atanacak
bulb.pos.z	-	-	\$body.height+\$base.height	Gövde yüksekliği ile taban yüksekliğinin toplam değeri atanacak
body.width	3	10	-	
body.length	3	10	-	
body.height	20	40	-	
body.pos.x	-	-	random (\$base.min.x+\$body.width/2) (\$base.max.x-\$body.width/2)	Taban genişliği içerisinde kalacak rastgele bir pozisyon değeri atanacak
body.pos.y	-	-	random (\$base.max.y-\$body.length/2) (\$base.min.y+\$body.length/2)	Taban uzunluğu içerisinde kalacak rastgele bir pozisyon değeri atanacak
body.pos.z			\$base.height	Taban yüksekliği atanarak tabanın üzerinde olması sağlanacak
base.width	10	30	-	
base.length	10	30	-	
base.height	2	5	-	
base.pos.x	-	-	-	Varsayılan değer 0'dır
base.pos.y	-	-	-	Varsayılan değer 0'dır
base.pos.z	-	-	-	Varsayılan değer 0'dır

Çizelge 5-4 Uygulanan 3ds Max deęiřtiricileri için verilen sınır deęerler

Modifier Parametreleri	Min.	Max.	Uygulandıęı Birimler
Spherify.amount	0	20	head
Bend.angle	0	20	head
Skew.amount	0	5	Head, body
Taper.amount	0	-1	head
Taper.lowerLimit	0	20	head
Taper.upperLimit	30	50	head



Şekil 5-2 ConGen'in türettięi alternatif tasarımlardan örnekler

Şekil 5-2'deki alternatif türetimler incelendięinde, özellikle 3ds Max deęiřtiricilerinin uygulandıęı örneklerin, alışıldık olmayan formlara sahip olduęu görülmektedir. Deęiřtirici uygulanmayan alternatifler ise günümüz tasarım örneklerine çok yakın bir stil ailesinin ConGen aracılıęıyla kolaylıkla oluşturulabileceęini göstermektedirler.

5.3 Mimari Şematik Tasarım Örneği: Bir Konut Planlama Çalışması

Bu çalışmada beş farklı mekândan oluşan bir konut örneği ele alınmıştır. Bu mekânlar için: a) Salon (livingR): 30m² b)Mutfak (kitchen): 10m² c) Wc: 3m² d)Giriş (entrance): 5m² e) Yatak Odası (bedroom): 12m²'den oluşan alan sınırlamaları belirlenmiştir. Tüm alternatiflerin toplam alanları 60m²'dir. Ve hepsinde, Salon kuzeyde kalacak şekilde türetim yapılmıştır.

Mekânlara atanan renkler:

- 1- Salon: turuncu
- 2- Mutfak: açık yeşil
- 3- WC: açık mavi
- 4- Giriş: sarı
- 5- Yatak Odası: bordo

Çizelge 5-5 Konut türetimi için kullanılan değer ve hedef kodlar tablosu

Alt-birim parametreleri	Min.	Max.	Target	Kod Anlamı
livingR.width	400	700	-	
livingR.length			$300000/\$livingR.width$	Sınır alan değeri, türetilen genişlik değerine bölünerek gerekli uzunluk değeri bulunacaktır.
livingR.height	300	500	-	
livingR.pos.x	-	-	-	Varsayılan değer 0'dır
livingR.pos.y	-	-	-	Varsayılan değer 0'dır
livingR.pos.z	-	-	-	Varsayılan değer 0'dır
kitchen.width	250	400	-	
kitchen.length			$100000/\$kitchen.width$	Sınır alan değeri, türetilen genişlik değerine bölünerek gerekli uzunluk değeri bulunacaktır.
kitchen.height	300	400	-	
kitchen.pos.x	-	-	$random(\$livingR.max.x-\$kitchen.width/2)$ $(\$livingR.min.x+\$kitchen.width/2)$	Mutfağın x pozisyonu salonun genişliğinden taşmayacak şekilde rastgele belirlenecektir.
kitchen.pos.y	-	-	$\$livingR.min.y-\$kitchen.length/2$	Mutfağın y pozisyonu salonun minimum pozisyon değerini alacaktır.
kitchen.pos.z	-	-	-	Varsayılan değer 0'dır

wc.width	150	250	-	
wc.length			$30000/\$wc.width$	Sınır alan değeri, türetilen genişlik değerine bölünerek gerekli uzunluk değeri bulunacaktır.
wc.height	250	250	-	
wc.pos.x	-	-	if ($\$kitchen.max.x > \$livingR.max.x/2$) then $\$wc.pos.x = \$kitchen.min.x + \$wc.width/2$ else $\$wc.pos.x = \$kitchen.max.x - \$wc.width/2$	Mutfak salonun simetri aksının hangi yanındaysa wc'de aynı yanında olacak, x konumu mutfakla aynı olacaktır.
wc.pos.y	-	-	$\$kitchen.min.y - \$wc.length/2$	Mutfağın minimum y pozisyonu atanacaktır.
wc.pos.z	-	-	-	Varsayılan değer 0'dır
entrance.width	150	200	-	
entrance.length			$50000/\$entrance.width$	Sınır alan değeri, türetilen genişlik değerine bölünerek gerekli uzunluk değeri bulunacaktır.
entrance.height	300	300		
entrance.pos.x	-	-	if ($\$kitchen.max.x > \$livingR.max.x/2$) then $\$entrance.pos.x = \$kitchen.min.x - \$entrance.width/2$ else $\$entrance.pos.x = \$kitchen.max.x + \$entrance.width/2$	Mutfak salonun simetri aksının hangi yanındaysa giriş diğer yanında bulunacak ve mutfağa bitişik olacaktır.
entrance.pos.y	-	-	$\$livingR.min.y - \$entrance.length/2$	Salonun minimum y pozisyonu atanacaktır.
entrance.pos.z	-	-	-	Varsayılan değer 0'dır
bedroom.width	300	500		
bedroom.length			$120000/\$bedroom.width$	Sınır alan değeri, türetilen genişlik değerine bölünerek gerekli uzunluk değeri bulunacaktır.
bedroom.height	300	300		
bedroom.pos.x			if $\$kitchen.max.x > \$livingR.max.x/2$ then $\$bedroom.pos.x = \$entrance.max.x - \$bedroom.width/2$ else $\$bedroom.pos.x = \$entrance.min.x + \$bedroom.width/2$	Mutfak salonun simetri aksının hangi yanındaysa yatak odası diğer yanında bulunacak ve girişe bitişik olacaktır.
bedroom.pos.y			$\$entrance.min.y - \$bedroom.length/2$	Girişin minimum y pozisyonu atanacaktır.
bedroom.pos.z	-	-	-	Varsayılan değer 0'dır



Şekil 5-3 ConGen’de türetilen, hepsi aynı mekân büyüklüklerine ve toplam alana sahip şematik konut alternatifleri

Geliştirilen alternatiflerin ışığında, ConGen’in tasarım kısıtlamalarının denenmesi, belirli kısıtlara bağlı bir alternatif ailesinin türetilmesinde kesin bir başarıya sahip olduğu görülmektedir. Aynı kısıtlamalara sahip bu sayıda alternatifin bir tasarımcı tarafından oluşturulması çok uzun bir zaman gerektirirken, ConGen sayesinde kısa bir zaman kullanılarak istenilen çözüm alternatifleri geliştirilebilmiştir.

5.4 Önerilen Metodolojinin Kullanımı: Bir koltuk tasarımı

Bu örnekte, önerilen tasarım metodolojisinin uygulanması ele alınacaktır. Tasarım konusu olarak, tasarım tarihi boyunca çok farklı çözümlerin geliştirdiği, mimarlardan grafikerlere çok farklı tasarım disiplinlerinden tasarımcıların örnekler verdiği, bir oturma elemanı seçilmiştir.

1.- Varolan tasarım örneklerinin incelenmesi

Seçilen konu için, günümüzün ticari ürünlerinden, rastgele örnekler seçilmiştir. Bu örnekler

değerlendirilerek, hangi noktalarda ortak çözümlerin kullanıldığı, hangi noktadaki kararların farklılıkları oluşturduğu tespit edilmiştir. Şekil 6-4'te görülen tüm koltuklarda şu alt-birimler ayırt edilebilmektedir: a) Baza (base) b) Minder (cushion) c) İki adet (armA ve armB) d) Sırtlık (back) e) Ayaklar.



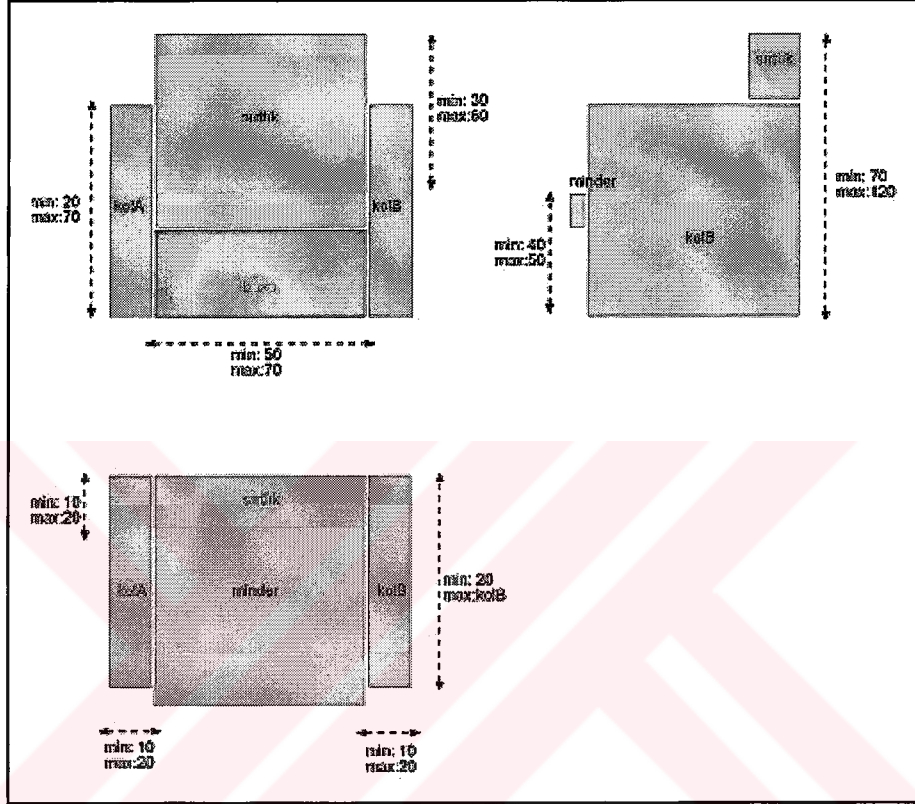
Şekil 5-4 Farklı markalara ait koltuk tasarımlarından örnekler

2.- Limitlerin ve alt-birimler arası ilişkilerin belirlenmesi

Bu aşama, ConGen'in türetim yapmak için kullanacağı temel bilginin aktarılmasını gerektirir. Alt-birimlerin boyutları ve pozisyonları ile ilgili bilgiler, analiz edilen örneklerdeki değerler karşılaştırılarak elde edilir. Ancak bu değerler nihai değildir, türetim aşamasında, belirli parametrelerin sınır değerlerini değiştirerek, deneme-yanılma yöntemiyle tasarımcı istediği sonuçları elde edebilir.

Örnekler arasında, kullanılan malzeme dışında, farklılık yaratan tercihlerin başında, alt-birimlerin boyutlarındaki farklılıklar gelmektedir. İkinci önemli tercih ise, alt-birimlerin boyutsal ve pozisyonel ilişkilerindedir. Örneğin baza ve minder ilişkisi, çoğu örnekte farklı çözülmüştür. Minder, bazadan dışarı taşarak ayrılabilir, aynı boyutlara sahip olup bir çizgiyle ayrıştırılabilir, ya da baza ile birleşip yokmuş gibi algılatılabilir. Aynı şekilde sırtlık ve kollar

uygun pozisyon ve boyutlara sahip olduklarında, birleşerek tek bir elemana dönüşmüşlerdir. Ayaklar her koltukta farklı malzeme ve boyutlarla kullanılmıştır. Ortak bir ölçüt çıkarılamayacağı için, koltuk ayaklarının çözümü detaylandırma aşamasına bırakılmıştır.



Şekil 5-5 Örneklerin değerlendirilmesiyle elde edilen sınır değerler

Çizelge 5-6 Koltuk türetimi için kullanılan değer ve hedef kodlar tablosu

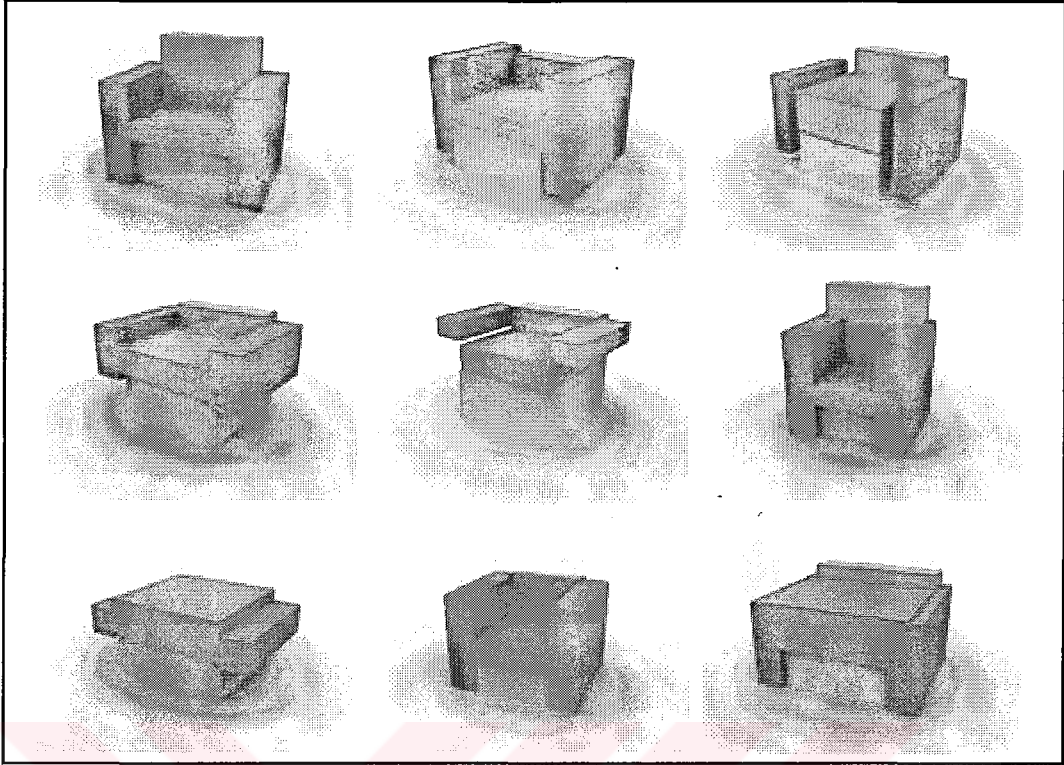
Alt-birim parametreleri	Min.	Max.	Target	Kod Anlamı
base.width	50	70	-	-
base.length	50	70	-	-
base.height	30	40	-	-
base.pos.x	-	-	-	Varsayılan değer 0'dır
base.pos.y	-	-	-	Varsayılan değer 0'dır
base.pos.z	-	-	-	Varsayılan değer 0'dır
cushion.width	-	-	\$base.width	Baza genişliği atanacaktır.

cushion.length	-	-	$\$base.length+10$	Baza uzunluğundan on fazla olacaktır.
cushion.height	-	-	$45-\$base.height$	Kırkbeş cm.den baza yüksekliği çıkarıldığında kalan değer atanacaktır.
cushion.pos.x	-	-	$\$base.pos.x$	Baza x konumu atanacaktır
cushion.pos.y	-	-	$\$base.pos.y$	Baza y konumu atanacaktır
cushion.pos.z	-	-	$\$base.max.z$	Bazanın en yüksek z konumu atanacaktır.
armA.width	10	20	-	
armA.length			random 20 cushion.length	Yirmi cm ile minder uzunluğu arasında rastgele bir değer atanacaktır.
armA.height			random 10 ($\$base.height + \$cushion.height+25$)	On cm. ile baza ve minder yüksekliği toplamından yirmibeş cm. fazla olan değer arasında rastgele belirlenecektir.
armA.pos.x	-	-	$\$base.max.x+\$armA.width/2$	Bazanın maksimum x konumu atanacaktır.
armA.pos.y			$\$base.pos.y$	Bazanın y konumu atanacaktır.
armA.pos.z	-	-	random 0 ($\$cushion.max.z+15$)	Sıfır ile minder maksimum z konumundan onbeş cm. fazla olan değer arasında rastgele belirlenecektir.
armB.width			$\$armA.width$	KolA genişliği atanacaktır.
armB.length			$\$armA.length$	KolA uzunluğu atanacaktır.
armB.height			$\$armA.height$	KolA yüksekliği atanacaktır.
armB.pos.x	-	-	$\$base.min.x-\$armB.width/2$	Bazanın minimum x konumu atanacaktır.
armB.pos.y	-	-	$\$armA.pos.y$	KolA y konumu atanacaktır.
armB.pos.z	-	-	$\$armA.pos.z$	KolA z konumu atanacaktır.
back.width	-	-	$\$cushion.width$	Minder genişliği atanacaktır.
back.length	10	20		
back.height			random ($\$base.height + \$cushion.height + 15$) 60	Baza ve minder toplam yüksekliğinden onbeş cm. fazla bir değerle altmış cm. arasında rastgele belirlenecektir.
back.pos.x			$\$base.pos.x$	Bazanın x konumu atanacaktır

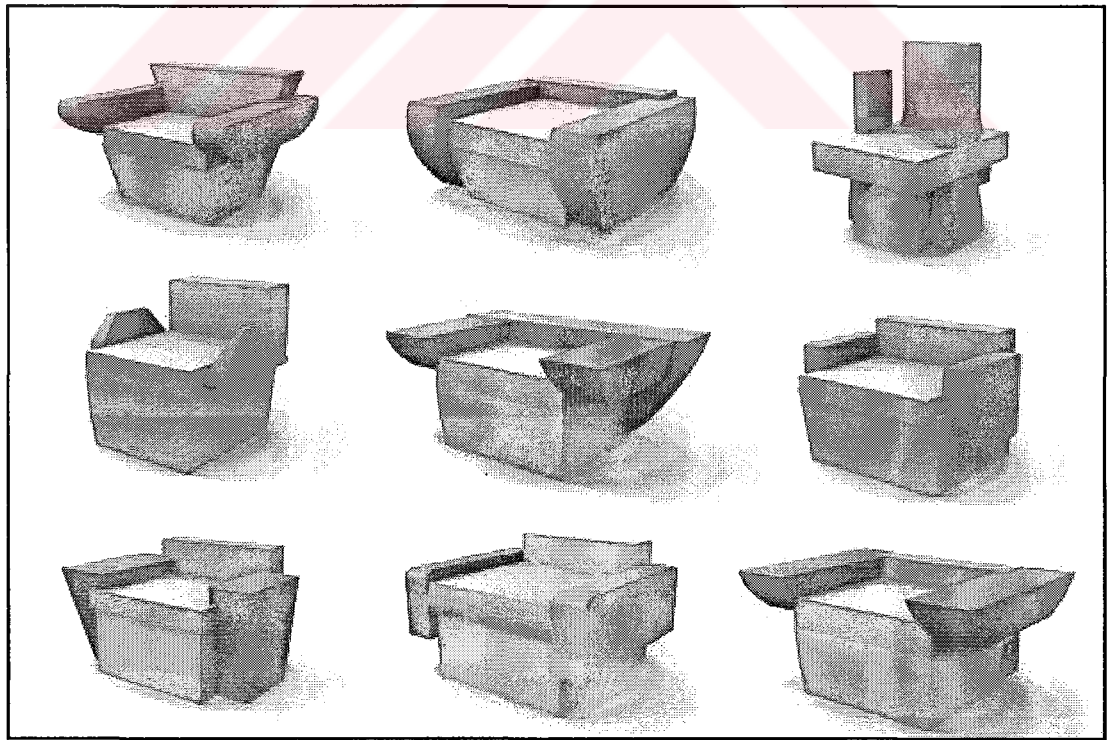
back.pos.y			$\$base.max.y+\$back.length/2$	Bazanın maksimum y konumu atanacaktır.
back.pos.z	-	-	-	Varsayılan değer 0 dir.

3.- Form Türetimi:

Alt-birim parametreleri için belirlenen limitlerin program arayüzüne aktarılmasından sonra türetim aşamasına geçilmektedir. ConGen’de türetim aşamaları üçe bölünmüştür. Kullanıcı boyutsal varyasyonları görmek amacıyla, boyutlandırma panelinde “Generate” düğmesini kullanarak türetimi başlatır. Daha sonra alt-birim pozisyonlarından doğabilecek alternatifleri türetebilmek için konumlandırma panelindeki “Generate” düğmesi kullanılır. Son olarak, ConGen’e alınmış Max değiştiricilerinin, tasarım üzerindeki etkilerini denemek için değiştiriciler panelindeki “Generate” düğmesi kullanılır. Türetme işlemi her başlatıldığında tek bir alternatifin türetilmesi tercih edilmiştir. Max programı içerisinde istenilen nesnelere kopyalanması kolaylıkla sağlanır. Kullanıcı, daha sonra karşılaştırmak isteyebileceği, dikkate değer bulduğu tasarım alternatiflerini kopyalayarak, türetmeye devam eder. Kopyalanmış olan nesnelere aktif nesnelere üzerindeki değişikliklerden etkilenmeyeceklerdir.



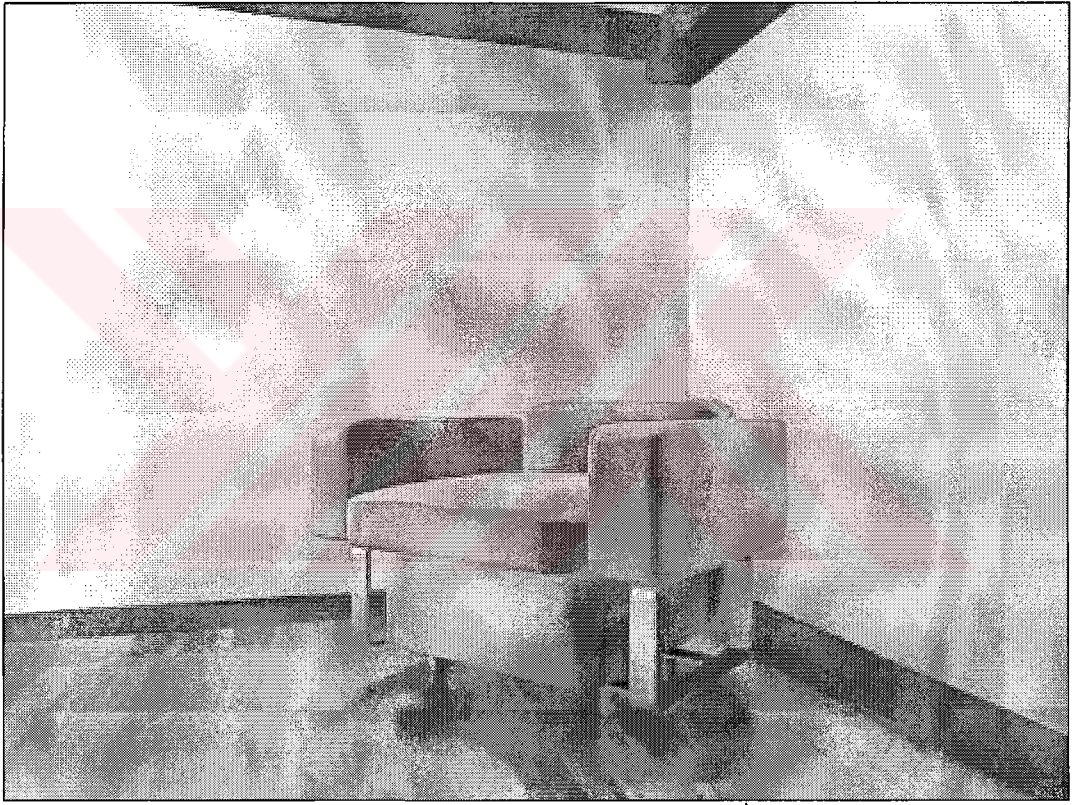
Şekil 5-6 Boyut ve pozisyon değerlerine dayanarak türetilen tasarım alternatiflerinden örnekler



Şekil 5-7 Max değıştiricilerinin uygulanmasıyla türetilebilecek alternatiflerden örnekler

4. ve 5.- Form Seçimi ve Detaylandırma

Yeterli sayıda form alternatifi üretildikten sonra, form seçimi, kullanıcı ihtiyaçlarına, malzeme kriterlerine ya da uzman görüşlerine dayanılarak yapılabilir. 3d Max programının özellikleri kullanılarak, gerçekçi çizimler üretilerek türetilen formların değerlendirilmesi kolaylaştırılabilir. Sunulan örnekte, ConGen'in türettiği koltuk alternatiflerinden biri seçilerek, malzeme ve renk gibi özellikleri geliştirilmiş, gerekli strüktürel eklemeler yapılarak tasarım süreci tamamlanmıştır.



Şekil 5-8 Final tasarım sunumu örneği

6. SONUÇLAR

Tez çalışmasının sonuçları iki bölümde ele alınacaklardır. Birinci bölümde, bir üretici sistem örneği olarak geliştirilen ConGen'in kullanımından elde edilen sonuçlar ve programın geleceği tartışılacak, ikinci bölümde ise ConGen ve incelenen diğer üretici sistem örneklerinin ışığında bu tür sistemlerin geleceği ve tasarım dünyasında oynayabilecekleri rol üzerine geliştirilen fikirler tartışılacaktır.

6.1 ConGen'in Kullanımı ve Geleceği

ConGen programı, üretici sistemlerin erken tasarım aşamalarında kullanımlarının tasarıma katkılarını tartışabilmek amacıyla geliştirilmiştir. Geliştirilmiş olan tasarım çözümleri, kesin ya da en uygun sonuçlar değildir. Ve üretilen çözümler, parametrik sınırların oluşturduğu genel çözümün etrafındaki kısıtlı alternatif uzayında bulunurlar. Bu olgu ilk bakışta, bu tür bir aracın tasarım yaratıcılığına katkısının olamayacağını düşündürmektedir. Ancak kullanım aşamasında yapılan, genel kabule göre hata sayılabilecek parametrik değişikliklerin ilham verici sonuçlar ürettiği gözlenmiştir. ConGen'in tasarım sürecinin erken aşamalarında kullanımı sırasında gözlemlenen faydaları şu maddelerle özetlenebilir:

- Tasarımcı, belirli bir tasarım problemi için çözüm geliştirirken kullandığı boyutsal, konumsal sınırlamaların etkilerini hızlı ve yanılmaz bir şekilde test edebilmektedir.
- Alışılmadık tasarım çözümleri geliştirebilmek için, alışılmamış nesne ilişkilerinin ve parametrelerin denenmesi, yaratıcı çözümlerin geliştirilmesine yardımcı olabilmektedir.
- ConGen, tasarımcı, müşteri, malzeme üreticisi ya da mühendis gibi tasarım kısıtlamaları belirleyen tarafların, belirledikleri kısıtlamaların etkilerini tartışabileceği bir iletişim aracı olarak kullanılabilir.

ConGen ile ilgili iki farklı geliştirme senaryosu bulunmaktadır. Programın, platform bağımsız ve ücretsiz olarak kullanılabilmesi için, farklı bir yazılım dilinde (örneğin Java3D) yeniden oluşturulması mümkün olabilir. Bu durumda 3ds Max araçlarından faydalanılamayacaktır. Ancak bu eksiklik, erken tasarım aşamalarında, kullanıcının daha fazla ihtiyaç duyabileceği ek araçların ve özel geometri sınıflarının oluşturulmasıyla giderilebilir. İkinci geliştirme senaryosu ise ConGen'in 3ds Max ile uyumluluğu artırılarak, şu anda kullanılamayan bazı Max özelliklerinin ConGen aracılığıyla kullanılabilmesini sağlamaktır. Öncelikli olarak eklenebilecek özellikler şu başlıklar altında toplanabilir:

- Alt-birim sayısındaki kısıtlama kaldırılarak, kullanıcının istediği sayıda alt-birim oluşturması
- 3ds Max animasyon özelliklerinin kullanılabilmesiyle, türetilen formlar arasındaki değişimin hareketli olarak izlenilebilmesi
- 3ds Max içerisinde bulunan hareket kontrol araçlarının ConGen ile ilişkilendirilerek kullanıcının sahne üzerinde yaptığı değişikliklerin gerçek zamanlı olarak, ilişkili nesnelerin boyut ve pozisyonlarına etkisinin gözlemlenebilmesi
- Alt-birim parametrelerine malzeme detayları eklenerek, bu konuda da alternatiflerin türetilmesi

ConGen ve benzeri araçların gelişme süreçleri, bu araçların tasarımcılar tarafından aktif kullanımıyla anlam kazanacaktır. Günümüz tasarımcılarının, her türlü önyargıdan kurtulup bilgisayarla ortaklıkta önemli bir yere sahip olan algoritmik düşünce şekline ayak uydurabilmeleri, bu alanda yapılacak çalışmalar için en önemli itici güç olacaktır.

6.2 Türetici Sistemlerin Tasarımda Kullanımı ve Gelecekleri

Tasarım türetimi deyimi, çoğu zaman, endüstriyel üretim yöntemlerini otomatize etmek, bir seferde çok sayıda ürünün üretimini sağlamak olarak anlaşılmaktadır. Hatta ortaya çıkmasının en büyük nedenlerinde biri de budur, ancak türetici sistemler sadece hatasız bir üretimi değil, bu seri üretimin monotonluğunun da kırılmasını sağlayabilir. Bu tezde örnekleri verilen türetici sistemlerin potansiyelleri konusunda birçok tanımlama yapılmıştır. Türetici sistemlerin potansiyelleri şu başlıklar altında toplanabilir.

1. Türetici sistemler bir tasarım ailesini ya da neslini tamamıyla türetebilmek açısından kesinlikle başarılıdır. Türetilecek tasarım örneğinin soyutlanma seviyesinin belirleyeceği şekilde, eldeki tasarımın potansiyelleri, üretim için tanımlanmış problem alanında tamamen anlaşılmış olur.
2. Bu tür sistemlerin, geçerli tasarım uzayının daha iyi araştırılabilmesini sağlayarak tasarımcının, yaratıcılığını arttırdığı söylenebilir. Ancak, bilgisayarlar ne için programlanmışlarsa sadece onu yapabilirler. Bu nedenle yaratıcılık geliştirme fikri ilk bakışta, hem yaratıcılık kavramının kendisinin sorunlu, tanımlanamaz doğasından hem de bu işi yapacak olanın bilgisayarlar olması nedeniyle anlamsız gelebilir. Ancak, türetici bir araç olmaksızın düşünülemez olan, çok sayıda tasarım elemanın

ilişkilerinden doğacak permutasyonların otomatik olarak elde edilebilmesinin, yeni fikirlere ve kavrayışlara ilham vereceği şüphesizdir.

3. Ele alınan örneklerde yer almayan, ancak üretici sistemlerle ilgili önemli bir başka iddia ise, bu tür araçların iyi tasarımları kötülerinden ayırt edebileceğidir. Bu iddia prensipte doğru olsa da ancak çok katı tanımlanmış problem alanlarında geçerli olabilir ki bu da tasarım alanında neredeyse imkânsızdır. Tasarım varyasyonları üretmek konusunda bilgisayarlar çok güçlüdür ancak bunları elemek için, ciddi bir bilgi birikimi ve ortak içgörü (common sense) gerekir. Ve bunları bir bilgisayara aktarabilmek çok zordur (Fischer, Herr 2001).

Türetici sistemlerin, tasarım sürecine yapabilecekleri katkıların tam olarak anlaşılabilmesi bu sistemlerin tasarım dünyasında yaygın olarak kullanılmaya başlanması ile mümkün olabilecektir. Bu aşamaya gelebilmek için, tasarımcı adaylarının, eğitim süreçlerinde sayısal üretim mantığıyla tanışmış olmaları, aktif tasarımcıların ise bilgisayarları ve tasarım programlarını, tasarım sunumunu hızlandırabilecek otomatik araçlar olarak görmekten vazgeçip, fikir geliştirme aşamasında da kullanılacak üreticiler olarak görebilmeleri gerekmektedir.

KAYNAKLAR

- Arnheim, R. (1969), *Visual Thinking*, University of California Press, Berkeley and Los Angeles.
- Candy, A. ve Edmonds, E. A. (1997), "Supporting the creative user: a criteria-based approach to interaction design", *Design Studies* 18, 185-194.
- Celani, G. (2001), "An educational experiment with shape grammars and computer applications", *International Journal of Design Computing*.
- Cross, N. (1997), "Descriptive models of Creative Design: Application to an example", *Design Studies* Vol. 18, No. 4, s. 427-440.
- Dejong, G. (1989), "The role of explanation in analogy; or, the curse of an alluring name", Vosniadou, S. ve Ortony, A. (ed.) "Similarity and analogical reasoning" içinde, s 346-365. Cambridge: Cambridge University Press.
- Do, E. Y. ve Gross, M. D. (1995), "Drawing Analogies: finding visual references by sketching", *Proceedings of Association of Computer Aided Design In Architecture (ACADIA)*, Seattle WA , 35-52.
- Eckert, C., Kelly, I., Stacey, M. (1999), "Interactive generative systems for conceptual design: An empirical perspective", *Artificial Intelligence for Engineering Design, Analysis and Manufacturing* , 13, 303–320.
- Eggink, D. A., Gross, M. D., Do, E. Y. (2001) "Smart Objects: Constraints and Behaviors in a 3D Design Environment", *ECAADE 2001 Conference Paper*.
- Eggink, D. A. (2001), "Smart Objects: Constraints and Behaviors in a Three-Dimensional Design Environment", *Master of Architecture Thesis, Department of Architecture, University of Washington 2001*.
- Fischer, T., Herr, C. M. (2001), "Teaching Generative Design", *Generative Art Conference 2001*
- Fischer, T. ve Fischer, T. (2003), "Toolmaking for Digital Morphogenesis", *International Journal of Design Computing, Volume:6 2003*.
- Gardner, B. M. (1997), "The Grid Sketcher – An AutoCad Based Tool For Conceptual Design Processes", *Msc. Thesis, Department of Architecture, University of Nevada*.
- Gentner, D. (1983), "Structure-mapping: a theoretical framework for analogy", *Cognitive Science*, 7, s 155-170.
- Getzels, J. W. (1964), "Creative thinking, problem-solving, and instruction.", *Theories of learning and instruction*, Hilgard E. R. (Derl.), 240-267, IL: University of Chicago Press, Chicago.
- Gero, J. S. (1983), "Computer-Aided Architectural Design - Past, Present and Future." *Architectural Science Review* 26(1): 2-5.
- Gero, J. S. (1995), "Computers and Creative Design", *SAFA, Architects and Computers*, Espoo, 13 p.
- Goel, V. (1995), *Sketches of Thought*, MIT Press, Cambridge, MA

- Goldschmidt, G. (1991), "The Dialectics of Sketching", *Creativity Research Journal*. Vol.4, No. 2, s.123-143
- Gross, M. D. (1989), "Relational Modeling: A Basis for Computer-Assisted Design", *The Electronic Design Studio*, MIT Press, Cambridge, Massachusetts, 123-136.
- Hemberg, M. ve O'Reilly, U. (2004), "Extending Grammatical Evolution to Evolve Digital Surfaces with Genr8", *Euro GP 2004*.
- Knight, T. (1999), "Shape Grammars in Education and Practice: History and Prospect", *International Journal of Design Computing*, Volume 2.
- Koning, H. ve Eisenberg, J. (1981), "The language of the prairie: Frank Lloyd Wright's prairie houses," *Environment and Planning B* 8 (1981): 295-323.
- Kwon, D. Y. (2003), "ArchiDNA: A Generative System for Shape Configuration", *Master of Science Thesis in Architecture University of Washington*.
- McGill, M. (2001), "A Visual Approach for Exploring Computational Design", *SMArchS thesis, MIT*.
- McGinty, T. (1979). *Design and the Design Process*. J. C. Snyder and A. J. Catanese (Ed.), "Introduction to Architecture" içinde (s. 152-190). New York: McGraw-Hill.
- Novick, L. (1988) "Analogical transfer, problem similarity, and expertise", *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 14(3), s 510-520.
- Oxman, R. (2002) "The thinking eye:visual re-cognition in design emergence", *Design Studies* 23, 135-164
- Pak, B. (2003), "A design model at the intersection of digital medium and architecture.", *Msc Thesis in Architectural Design Computing, ITU*
- Pak, B., Özener, O., Erdem, A., (2003), "Xp-GEN. A randomized design tool for non-deterministic digital design methods in architecture and visual design", *eCAADe 2003, Vienna*
- Scheurer, F. (2003), "The Groningen Twister An experiment in applied generative design", *Generative Art Conference 2003*.
- Schön, D. A.ve Wiggins, G. A. (1992), "Kinds of seeing and their function in designing", *Design Studies* 13(2), 135-156
- Stiny, G. (1976), "Two exercises in formal composition", *Environment and Planning B* 3:187-210.
- Stiny, G. ve Mitchell, W. (1978), "The Palladian grammar," *Environment and Planning B* 5 (1978): 5-18.
- Stiny, G. (2001), "How to calculate with shapes", *working paper, Department of Architecture, Design Computation, MIT, MA*.
- Soddu, C. (2002) "New Naturality: A Generative Approach to Art and Design", *LEONARDO*, Vol. 35, No. 3, 291-294 MIT Press
- Soufi, B. ve Edmonds, E. (1996) "The cognitive basis for emergence: implications for design support", *Design Studies* 17, 451-463.

Suwa, M., Gero, S. J. ve Purcell, T. (1999) "Unexpected discoveries: how designers discover hidden features in sketches", A workshop on visual and spatial reasoning in design, içinde, Gero, J. S. ve Tversky, B. (Derl.), MIT, MA.

Tversky, B. (1999), "What does drawing reveal about thinking?", Gero, J. S. ve Tversky, B. (Derl.), A workshop on visual spatial reasoning in design, MIT, MA.

Wang, Y. ve Duarte J.P. (2001), "Automatic Generation and Fabrication of Designs", Automation in Construction 447, Elsevier.

İNTERNET KAYNAKLARI

[1] www.autodesk.com (Revit programı üretici firması resmi web sayfası)

[2] www.celestinosoddu.com

[3] www.discreet.com (3ds Max programı üretici firması resmi web sayfası)

[4] www.generativedesign.com

[5] www.mit.edu/edgsrc/www/genr8/index.html (Emergent Design Group web sitesi)

[6] www.tdk.gov.tr/bilterim/ (Türk Dil Kurumu Bilgisayar Terimleri Karşılıklar Klavuzu)

EKLER

Ek 1 ConGen programının kaynak kodu.



Ek 1 ConGen programının kaynak kodu

```

macroScript ConGen category:"saros"(

global zone01
global zone02
global zone03
global zone04
global zone05
global z01_meshed=false
global z02_meshed=false
global z03_meshed=false
global z04_meshed=false
global z05_meshed=false

zone01=box lengthsegs:12 widthsegs:12 heightsegs:12
zone02=box lengthsegs:12 widthsegs:12 heightsegs:12
zone03=box lengthsegs:12 widthsegs:12 heightsegs:12
zone04=box lengthsegs:12 widthsegs:12 heightsegs:12
zone05=box lengthsegs:12 widthsegs:12 heightsegs:12

rollout zDimensions "Dimensions" width:330 height:1000
(

--ZONE DIMENSIONS INTERFACE SETTINGS

label z02_w_label "Width:" pos:[34,195] width:32 height:14
groupBox z02_group "Object02" pos:[7,149] width:330 height:123
button save "Save Settings" pos:[258,672] width:80 height:19 enabled:true
button load "Load Settings" pos:[170,672] width:80 height:19 enabled:true
button generate "Generate!!!" pos:[9,672] width:80 height:19 enabled:true
spinner z02_w_min "min." pos:[82,195] width:60 height:16 range:[0,9999,0]
spinner z02_w_max "max." pos:[167,195] width:60 height:16 range:[0,9999,0]
label z02_l_label "Length:" pos:[30,219] width:40 height:14
spinner z02_l_min "min." pos:[82,218] width:60 height:16 range:[0,9999,0]
spinner z02_l_max "max." pos:[167,218] width:60 height:16 range:[0,9999,0]
label z02_h_label "Height:" pos:[32,243] width:35 height:17
spinner z02_h_min "min." pos:[82,243] width:60 height:16 range:[0,9999,0]
spinner z02_h_max "max." pos:[167,242] width:60 height:16 range:[0,9999,0]
label z03_w_label "Width:" pos:[34,326] width:30 height:14
groupBox z03_group "Object03" pos:[7,281] width:330 height:120
spinner z03_w_min "min." pos:[82,326] width:60 height:16 range:[0,9999,0]
spinner z03_w_max "max." pos:[167,326] width:60 height:16 range:[0,9999,0]
label z03_l_label "Length:" pos:[30,352] width:40 height:14
spinner z03_l_min "min." pos:[82,350] width:60 height:16 range:[0,9999,0]
spinner z03_l_max "max." pos:[167,350] width:60 height:16 range:[0,9999,0]
label z03_h_label "Height:" pos:[33,373] width:37 height:17
spinner z03_h_min "min." pos:[82,374] width:60 height:16 range:[0,9999,0]

```

spinner z03_h_max "max." pos:[167,374] width:60 height:16 range:[0,9999,0]
 label z04_w_label "Width:" pos:[34,453] width:31 height:14
 groupBox z04_group "Object04" pos:[8,408] width:330 height:122
 spinner z04_w_min "min." pos:[82,453] width:60 height:16 range:[0,9999,0]
 spinner z04_w_max "max." pos:[167,453] width:60 height:16 range:[0,9999,0]
 label z04_l_label "Length:" pos:[30,477] width:40 height:14
 spinner z04_l_min "min." pos:[82,477] width:60 height:16 range:[0,9999,0]
 spinner z04_l_max "max." pos:[167,477] width:60 height:16 range:[0,9999,0]
 label z04_h_label "Height:" pos:[30,500] width:40 height:17
 spinner z04_h_min "min." pos:[82,501] width:60 height:16 range:[0,9999,0]
 spinner z04_h_max "max." pos:[167,501] width:60 height:16 range:[0,9999,0]
 label z05_w_label "Width:" pos:[37,586] width:30 height:14
 groupBox z05_group "Object05" pos:[8,539] width:330 height:120
 spinner z05_w_min "min." pos:[82,586] width:60 height:16 range:[0,9999,0]
 spinner z05_w_max "max." pos:[167,586] width:60 height:16 range:[0,9999,0]
 label z05_l_label "Length:" pos:[33,609] width:36 height:14
 spinner z05_l_min "min." pos:[82,609] width:60 height:16 range:[0,9999,0]
 spinner z05_l_max "max." pos:[167,609] width:60 height:16 range:[0,9999,0]
 label z05_h_label "Height:" pos:[34,630] width:36 height:17
 spinner z05_h_min "min." pos:[82,631] width:60 height:16 range:[0,9999,0]
 spinner z05_h_max "max." pos:[167,631] width:60 height:16 range:[0,9999,0]
 label z01_w_label "Width:" pos:[35,63] width:32 height:14
 groupBox z01_group "Object01" pos:[8,20] width:330 height:123
 spinner z01_w_min "min." pos:[82,63] width:60 height:16 range:[0,9999,0]
 spinner z01_w_max "max." pos:[167,63] width:60 height:16 range:[0,9999,0]
 label z01_l_label "Length:" pos:[31,88] width:39 height:14
 spinner z01_l_min "min." pos:[82,88] width:60 height:16 range:[0,9999,0]
 spinner z01_l_max "max." pos:[167,88] width:60 height:16 range:[0,9999,0]
 label z01_h_label "Height:" pos:[32,113] width:38 height:17
 spinner z01_h_min "min." pos:[82,113] width:60 height:16 range:[0,9999,0]
 spinner z01_h_max "max." pos:[167,113] width:60 height:16 range:[0,9999,0]
 editText z01_w_target "target" pos:[240,63] width:75 height:16
 editText z01_name "Name: " pos:[29,40] width:113 height:14
 editText z01_l_target "target" pos:[240,88] width:75 height:16
 editText z01_h_target "target" pos:[240,112] width:75 height:16
 editText z02_name "Name: " pos:[29,173] width:113 height:14
 editText z02_w_target "target" pos:[240,194] width:75 height:16
 editText z02_l_target "target" pos:[240,218] width:75 height:16
 editText z02_h_target "target" pos:[240,242] width:75 height:16
 editText z03_w_target "target" pos:[240,326] width:75 height:16
 editText z03_l_target "target" pos:[240,350] width:75 height:16
 editText z03_h_target "target" pos:[240,374] width:75 height:16
 editText z03_name "Name: " pos:[28,303] width:113 height:14
 editText z04_w_target "target" pos:[240,453] width:75 height:16
 editText z04_l_target "target" pos:[240,477] width:75 height:16
 editText z04_h_target "target" pos:[240,501] width:75 height:16
 editText z04_name "Name: " pos:[28,430] width:113 height:14
 editText z05_name "Name: " pos:[29,564] width:113 height:14
 editText z05_w_target "target" pos:[240,586] width:75 height:16
 editText z05_l_target "target" pos:[240,609] width:75 height:16
 editText z05_h_target "target" pos:[240,631] width:75 height:16

```

button z01_w_target_b ">" pos:[318,63] width:13 height:16
button z01_l_target_b ">" pos:[318,88] width:13 height:16
button z01_h_target_b ">" pos:[318,114] width:13 height:16
button z02_w_target_b ">" pos:[319,192] width:13 height:16
button z02_l_target_b ">" pos:[319,217] width:13 height:16
button z02_h_target_b ">" pos:[319,242] width:13 height:16
button z03_w_target_b ">" pos:[318,326] width:13 height:16
button z03_l_target_b ">" pos:[318,350] width:13 height:16
button z03_h_target_b ">" pos:[318,374] width:13 height:16
button z04_w_target_b ">" pos:[320,454] width:13 height:16
button z04_l_target_b ">" pos:[320,478] width:13 height:16
button z04_h_target_b ">" pos:[320,502] width:13 height:16
button z05_w_target_b ">" pos:[319,586] width:13 height:16
button z05_l_target_b ">" pos:[319,609] width:13 height:16
button z05_h_target_b ">" pos:[319,631] width:13 height:16
colorPicker z01_color "Color:" pos:[156,39] width:71 height:16 color:(color 0 0 155)
colorPicker z02_color "Color:" pos:[155,172] width:71 height:16 color:(color 0 0 155)
colorPicker z03_color "Color:" pos:[156,303] width:71 height:16 color:(color 0 0 155)
colorPicker z04_color "Color:" pos:[156,429] width:71 height:16 color:(color 0 0 155)
colorPicker z05_color "Color:" pos:[155,564] width:71 height:16 color:(color 0 0 155)
button z01_select "Selection" pos:[282,40] width:50 height:18
button z02_select "Selection" pos:[282,170] width:50 height:18
button z03_select "Selection" pos:[281,302] width:50 height:18
button z04_select "Selection" pos:[283,431] width:50 height:18
button z05_select "Selection" pos:[283,563] width:50 height:18

```

--WORKS AFTER GENERATE! PRESSED

on generate pressed do(

--ZONE VALIDATION AND RECREATION

```

if (IsValidNode zone01)==false then zone01=box lengthsegs:12 widthsegs:12 heightsegs:12
if (IsValidNode zone02)==false then zone02=box lengthsegs:12 widthsegs:12 heightsegs:12
if (IsValidNode zone03)==false then zone03=box lengthsegs:12 widthsegs:12 heightsegs:12
if (IsValidNode zone04)==false then zone04=box lengthsegs:12 widthsegs:12 heightsegs:12
if (IsValidNode zone05)==false then zone05=box lengthsegs:12 widthsegs:12 heightsegs:12

```

--DIMENSION LIMITS FOR ZONE01

```

if z01_meshed==false then(
zone01.width=random z01_w_min.value z01_w_max.value
zone01.length=random z01_l_min.value z01_l_max.value
zone01.height=random z01_h_min.value z01_h_max.value
)
else(
zone01.scale.x=random (z01_w_min.value*zone01.scale.x/(zone01.max.x-zone01.min.x))
(z01_w_max.value*zone01.scale.x/(zone01.max.x-zone01.min.x))
zone01.scale.y=random (z01_l_min.value*zone01.scale.y/(zone01.max.y-zone01.min.y))
(z01_l_max.value*zone01.scale.y/(zone01.max.y-zone01.min.y))

```

```

zone01.scale.z=random (z01_h_min.value*zone01.scale.z/(zone01.max.z-zone01.min.z))
(z01_h_max.value*zone01.scale.z/(zone01.max.z-zone01.min.z))
)

```

```

if z01_name.text!="" then zone01.name=z01_name.text
zone01.material=standard diffuse:orange

```

--DIMENSION LIMITS FOR ZONE02

```

if z02_meshed==false then(
zone02.width=random z02_w_min.value z02_w_max.value
zone02.length=random z02_l_min.value z02_l_max.value
zone02.height=random z02_h_min.value z02_h_max.value
)
else(
zone02.scale.x=random (z02_w_min.value*zone02.scale.x/(zone02.max.x-zone02.min.x))
(z02_w_max.value*zone02.scale.x/(zone02.max.x-zone02.min.x))
zone02.scale.y=random (z02_l_min.value*zone02.scale.y/(zone02.max.y-zone02.min.y))
(z02_l_max.value*zone02.scale.y/(zone02.max.y-zone02.min.y))
zone02.scale.z=random (z02_h_min.value*zone02.scale.z/(zone02.max.z-zone02.min.z))
(z02_h_max.value*zone02.scale.z/(zone02.max.z-zone02.min.z))
)

```

```

if z02_name.text!="" then zone02.name=z02_name.text
zone02.material=standard diffuse:orange

```

--DIMENSION LIMITS FOR ZONE03

```

if z03_meshed==false then(
zone03.width=random z03_w_min.value z03_w_max.value
zone03.length=random z03_l_min.value z03_l_max.value
zone03.height=random z03_h_min.value z03_h_max.value
)
else(
zone03.scale.x=random (z03_w_min.value*zone03.scale.x/(zone03.max.x-zone03.min.x))
(z03_w_max.value*zone03.scale.x/(zone03.max.x-zone03.min.x))
zone03.scale.y=random (z03_l_min.value*zone03.scale.y/(zone03.max.y-zone03.min.y))
(z03_l_max.value*zone03.scale.y/(zone03.max.y-zone03.min.y))
zone03.scale.z=random (z03_h_min.value*zone03.scale.z/(zone03.max.z-zone03.min.z))
(z03_h_max.value*zone03.scale.z/(zone03.max.z-zone03.min.z))
)

```

```

if z03_name.text!="" then zone03.name=z03_name.text
zone03.material=standard diffuse:orange

```

--DIMENSION LIMITS FOR ZONE04

```

if z04_meshed==false then(
zone04.width=random z04_w_min.value z04_w_max.value
zone04.length=random z04_l_min.value z04_l_max.value
zone04.height=random z04_h_min.value z04_h_max.value
)
else(
zone04.scale.x=random (z04_w_min.value*zone04.scale.x/(zone04.max.x-zone04.min.x))
(z04_w_max.value*zone04.scale.x/(zone04.max.x-zone04.min.x))
zone04.scale.y=random (z04_l_min.value*zone04.scale.y/(zone04.max.y-zone04.min.y))

```

```

        (z04_l_max.value*zone04.scale.y/(zone04.max.y-zone04.min.y))
zone04.scale.z=random (z04_h_min.value*zone04.scale.z/(zone04.max.z-zone04.min.z))
        (z04_h_max.value*zone04.scale.z/(zone04.max.z-zone04.min.z))
    )
if z04_name.text!="" then zone04.name=z04_name.text
zone04.material=standard diffuse:orange

```

--DIMENSION LIMITS FOR ZONE05

```

if z05_meshed==false then(
zone05.width=random z05_w_min.value z05_w_max.value
zone05.length=random z05_l_min.value z05_l_max.value
zone05.height=random z05_h_min.value z05_h_max.value
)
else(
zone05.scale.x=random (z05_w_min.value*zone05.scale.x/(zone05.max.x-zone05.min.x))
        (z05_w_max.value*zone05.scale.x/(zone05.max.x-zone05.min.x))
zone05.scale.y=random (z05_l_min.value*zone05.scale.y/(zone05.max.y-zone05.min.y))
        (z05_l_max.value*zone05.scale.y/(zone05.max.y-zone05.min.y))
zone05.scale.z=random (z05_h_min.value*zone05.scale.z/(zone05.max.z-zone05.min.z))
        (z05_h_max.value*zone05.scale.z/(zone05.max.z-zone05.min.z))
)
if z05_name.text!="" then zone05.name=z05_name.text
zone05.material=standard diffuse:orange

```

--TRACING TARGET COMMANDS

```

try(if z01_w_target.text!="" do zone01.width=execute z01_w_target.text)catch()
try(if z01_l_target.text!="" do zone01.length=execute z01_l_target.text)catch()
try(if z01_h_target.text!="" do zone01.height=execute z01_h_target.text)catch()

try(if z02_w_target.text!="" do zone02.width=execute z02_w_target.text)catch()
try(if z02_l_target.text!="" do zone02.length=execute z02_l_target.text)catch()
try(if z02_h_target.text!="" do zone02.height=execute z02_h_target.text)catch()

try(if z03_w_target.text!="" do zone03.width=execute z03_w_target.text)catch()
try(if z03_l_target.text!="" do zone03.length=execute z03_l_target.text)catch()
try(if z03_h_target.text!="" do zone03.height=execute z03_h_target.text)catch()

try(if z04_w_target.text!="" do zone04.width=execute z04_w_target.text)catch()
try(if z04_l_target.text!="" do zone04.length=execute z04_l_target.text)catch()
try(if z04_h_target.text!="" do zone04.height=execute z04_h_target.text)catch()

try(if z05_w_target.text!="" do zone05.width=execute z05_w_target.text)catch()
try(if z05_l_target.text!="" do zone05.length=execute z05_l_target.text)catch()
try(if z05_h_target.text!="" do zone05.height=execute z05_h_target.text)catch()

```

--ZONE NAMES UPDATE

```

if zone01.name!="" then z01_group.caption=zone01.name
if zone02.name!="" then z02_group.caption=zone02.name

```

```

if zone03.name!="" then z03_group.caption=zone03.name
if zone04.name!="" then z04_group.caption=zone04.name
if zone05.name!="" then z05_group.caption=zone05.name

```

```
--UPDATING ZONE COLORS
```

```

zone01.material.diffuse=z01_color.color
zone02.material.diffuse=z02_color.color
zone03.material.diffuse=z03_color.color
zone04.material.diffuse=z04_color.color
zone05.material.diffuse=z05_color.color

```

```
)--END ON GENERATE
```

```
--WORKS FOR TARGET BOXES
```

```

on z01_w_target_b pressed do(
  rollout tText "Target Width for Object01" width:400 height:300(
    editText targetText "Script:" pos:[5,5] width:350 height:200
    on tText open do(if z01_w_target.text!="" do targetText.text=z01_w_target.text)
    on tText close do(if targetText.text!="" do z01_w_target.text=targetText.text)
  )
  createDialog tText 380 220 true
)

```

```

on z01_l_target_b pressed do(
  rollout tText "Target Length for Object01" width:400 height:300(
    editText targetText "Script:" pos:[5,5] width:350 height:200
    on tText open do(if z01_l_target.text!="" do targetText.text=z01_l_target.text)
    on tText close do(if targetText.text!="" do z01_l_target.text=targetText.text)
  )
  createDialog tText 380 220 true
)

```

```

on z01_h_target_b pressed do(
  rollout tText "Target Height for Object01" width:400 height:300(
    editText targetText "Script:" pos:[5,5] width:350 height:200
    on tText open do(if z01_h_target.text!="" do targetText.text=z01_h_target.text)
    on tText close do(if targetText.text!="" do z01_h_target.text=targetText.text)
  )
  createDialog tText 380 220 true
)

```

```

on z02_w_target_b pressed do(
  rollout tText "Target Width for Object02" width:400 height:300(
    editText targetText "Script:" pos:[5,5] width:350 height:200
    on tText open do(if z02_w_target.text!="" do targetText.text=z02_w_target.text)
    on tText close do(if targetText.text!="" do z02_w_target.text=targetText.text)
  )
  createDialog tText 380 220 true
)

```

```

on z02_l_target_b pressed do(
  rollout tText "Target Length for Object02" width:400 height:300(
    editText targetText "Script:" pos:[5,5] width:350 height:200
    on tText open do(if z02_l_target.text!=" do targetText.text=z02_l_target.text)
    on tText close do(if targetText.text!=" do z02_l_target.text=targetText.text)
  )
  createDialog tText 380 220 true
)

```

```

on z02_h_target_b pressed do(
  rollout tText "Target Height for Object02" width:400 height:300(
    editText targetText "Script:" pos:[5,5] width:350 height:200
    on tText open do(if z02_h_target.text!=" do targetText.text=z02_h_target.text)
    on tText close do(if targetText.text!=" do z02_h_target.text=targetText.text)
  )
  createDialog tText 380 220 true
)

```

```

on z03_w_target_b pressed do(
  rollout tText "Target Width for Object03" width:400 height:300(
    editText targetText "Script:" pos:[5,5] width:350 height:200
    on tText open do(if z03_w_target.text!=" do targetText.text=z03_w_target.text)
    on tText close do(if targetText.text!=" do z03_w_target.text=targetText.text)
  )
  createDialog tText 380 220 true
)

```

```

on z03_l_target_b pressed do(
  rollout tText "Target Length for Object03" width:400 height:300(
    editText targetText "Script:" pos:[5,5] width:350 height:200
    on tText open do(if z03_l_target.text!=" do targetText.text=z03_l_target.text)
    on tText close do(if targetText.text!=" do z03_l_target.text=targetText.text)
  )
  createDialog tText 380 220 true
)

```

```

on z03_h_target_b pressed do(
  rollout tText "Target Height for Object03" width:400 height:300(
    editText targetText "Script:" pos:[5,5] width:350 height:200
    on tText open do(if z03_h_target.text!=" do targetText.text=z03_h_target.text)
    on tText close do(if targetText.text!=" do z03_h_target.text=targetText.text)
  )
  createDialog tText 380 220 true
)

```

```

on z04_w_target_b pressed do(
  rollout tText "Target Width for Object04" width:400 height:300(
    editText targetText "Script:" pos:[5,5] width:350 height:200
    on tText open do(if z04_w_target.text!=" do targetText.text=z04_w_target.text)
    on tText close do(if targetText.text!=" do z04_w_target.text=targetText.text)
  )

```

```
)
createDialog tText 380 220 true
)
```

```
on z04_l_target_b pressed do(
rollout tText "Target Length for Object04" width:400 height:300(
editText targetText "Script:" pos:[5,5] width:350 height:200
on tText open do(if z04_l_target.text!="" do targetText.text=z04_l_target.text)
on tText close do(if targetText.text!="" do z04_l_target.text=targetText.text)
)
createDialog tText 380 220 true
)
```

```
on z04_h_target_b pressed do(
rollout tText "Target Height for Object04" width:400 height:300(
editText targetText "Script:" pos:[5,5] width:350 height:200
on tText open do(if z04_h_target.text!="" do targetText.text=z04_h_target.text)
on tText close do(if targetText.text!="" do z04_h_target.text=targetText.text)
)
createDialog tText 380 220 true
)
```

```
on z05_w_target_b pressed do(
rollout tText "Target Width for Object05" width:400 height:300(
editText targetText "Script:" pos:[5,5] width:350 height:200
on tText open do(if z05_w_target.text!="" do targetText.text=z05_w_target.text)
on tText close do(if targetText.text!="" do z05_w_target.text=targetText.text)
)
createDialog tText 380 220 true
)
```

```
on z05_l_target_b pressed do(
rollout tText "Target Length for Object05" width:400 height:300(
editText targetText "Script:" pos:[5,5] width:350 height:200
on tText open do(if z05_l_target.text!="" do targetText.text=z05_l_target.text)
on tText close do(if targetText.text!="" do z05_l_target.text=targetText.text)
)
createDialog tText 380 220 true
)
```

```
on z05_h_target_b pressed do(
rollout tText "Target Height for Object05" width:400 height:300(
editText targetText "Script:" pos:[5,5] width:350 height:200
on tText open do(if z05_h_target.text!="" do targetText.text=z05_h_target.text)
on tText close do(if targetText.text!="" do z05_h_target.text=targetText.text)
)
createDialog tText 380 220 true
)
```

--ATTACHING SELECTED AS ZONE01

```
on z01_select pressed do(
z01_meshed=true
for o in selection do(
convertToMesh o
)
sel=selection as array
converttoMesh zone01
for i=1 to sel.count do(
attach zone01 sel[i]
)
)
```

)--END ON ZONE01 PICK ATTACH

--ATTACHING SELECTED AS ZONE02

```
on z02_select pressed do(
z02_meshed=true
for o in selection do(
convertToMesh o
)
sel=selection as array
converttoMesh zone02
for i=1 to sel.count do(
attach zone02 sel[i]
)
)
```

)--END ON ZONE02 PICK ATTACH

--ATTACHING SELECTED AS ZONE03

```
on z03_select pressed do(
z03_meshed=true
for o in selection do(
convertToMesh o
)
sel=selection as array
converttoMesh zone03
for i=1 to sel.count do(
attach zone03 sel[i]
)
)
```

)--END ON ZONE03 PICK ATTACH

--ATTACHING SELECTED AS ZONE04

```
on z04_select pressed do(
z04_meshed=true
for o in selection do(
convertToMesh o
)
sel=selection as array
```

```

converttoMesh zone04
for i=1 to sel.count do(
attach zone04 sel[i]
)
)--END ON ZONE04 PICK ATTACH

```

--ATTACHING SELECTED AS ZONE05

```

on z05_select pressed do(
z05_meshed=true
for o in selection do(
convertToMesh o
)
sel=selection as array
converttoMesh zone05
for i=1 to sel.count do(
attach zone05 sel[i]
)
)--END ON ZONE05 PICK ATTACH

```

--SAVE DIMENSION SETTINGS

```

on save pressed do(
fileIni=getSaveFileName caption:"Save as"
setINISetting fileIni "Zone01" "wMin" (z01_w_min.value as string)
setINISetting fileIni "Zone01" "wMax" (z01_w_max.value as string)
setINISetting fileIni "Zone01" "lMin" (z01_l_min.value as string)
setINISetting fileIni "Zone01" "lMax" (z01_l_max.value as string)
setINISetting fileIni "Zone01" "hMin" (z01_h_min.value as string)
setINISetting fileIni "Zone01" "hMax" (z01_h_max.value as string)
setINISetting fileIni "Zone01" "wTarget" (z01_w_target.text as string)
setINISetting fileIni "Zone01" "lTarget" (z01_l_target.text as string)
setINISetting fileIni "Zone01" "hTarget" (z01_h_target.text as string)
setINISetting fileIni "Zone01" "Name" (z01_name.text as string)
setINISetting fileIni "Zone02" "wMin" (z02_w_min.value as string)
setINISetting fileIni "Zone02" "wMax" (z02_w_max.value as string)
setINISetting fileIni "Zone02" "lMin" (z02_l_min.value as string)
setINISetting fileIni "Zone02" "lMax" (z02_l_max.value as string)
setINISetting fileIni "Zone02" "hMin" (z02_h_min.value as string)
setINISetting fileIni "Zone02" "hMax" (z02_h_max.value as string)
setINISetting fileIni "Zone02" "wTarget" (z02_w_target.text as string)
setINISetting fileIni "Zone02" "lTarget" (z02_l_target.text as string)
setINISetting fileIni "Zone02" "hTarget" (z02_h_target.text as string)
setINISetting fileIni "Zone02" "Name" (z02_name.text as string)
setINISetting fileIni "Zone03" "wMin" (z03_w_min.value as string)
setINISetting fileIni "Zone03" "wMax" (z03_w_max.value as string)
setINISetting fileIni "Zone03" "lMin" (z03_l_min.value as string)
setINISetting fileIni "Zone03" "lMax" (z03_l_max.value as string)
setINISetting fileIni "Zone03" "hMin" (z03_h_min.value as string)
setINISetting fileIni "Zone03" "hMax" (z03_h_max.value as string)
setINISetting fileIni "Zone03" "wTarget" (z03_w_target.text as string)

```

```

setINISetting fileIni "Zone03" "lTarget" (z03_l_target.text as string)
setINISetting fileIni "Zone03" "hTarget" (z03_h_target.text as string)
setINISetting fileIni "Zone03" "Name" (z03_name.text as string)
setINISetting fileIni "Zone04" "wMin" (z04_w_min.value as string)
setINISetting fileIni "Zone04" "wMax" (z04_w_max.value as string)
setINISetting fileIni "Zone04" "lMin" (z04_l_min.value as string)
setINISetting fileIni "Zone04" "lMax" (z04_l_max.value as string)
setINISetting fileIni "Zone04" "hMin" (z04_h_min.value as string)
setINISetting fileIni "Zone04" "hMax" (z04_h_max.value as string)
setINISetting fileIni "Zone04" "wTarget" (z04_w_target.text as string)
setINISetting fileIni "Zone04" "lTarget" (z04_l_target.text as string)
setINISetting fileIni "Zone04" "hTarget" (z04_h_target.text as string)
setINISetting fileIni "Zone04" "Name" (z04_name.text as string)
setINISetting fileIni "Zone05" "wMin" (z05_w_min.value as string)
setINISetting fileIni "Zone05" "wMax" (z05_w_max.value as string)
setINISetting fileIni "Zone05" "lMin" (z05_l_min.value as string)
setINISetting fileIni "Zone05" "lMax" (z05_l_max.value as string)
setINISetting fileIni "Zone05" "hMin" (z05_h_min.value as string)
setINISetting fileIni "Zone05" "hMax" (z05_h_max.value as string)
setINISetting fileIni "Zone05" "wTarget" (z05_w_target.text as string)
setINISetting fileIni "Zone05" "lTarget" (z05_l_target.text as string)
setINISetting fileIni "Zone05" "hTarget" (z05_h_target.text as string)
setINISetting fileIni "Zone05" "Name" (z05_name.text as string)

```

)--end on save dimensions settings

--LOAD DIMENSION SETTINGS

```

on load pressed do(
fileIni=getOpenFileName caption:"Load"

```

```

try(z01_w_min.value = (getINISetting fileIni "Zone01" "wMin") as integer)catch()
try(z01_w_max.value = (getINISetting fileIni "Zone01" "wMax") as integer)catch()
try(z01_l_min.value = (getINISetting fileIni "Zone01" "lMin") as integer)catch()
try(z01_l_max.value = (getINISetting fileIni "Zone01" "lMax") as integer)catch()
try(z01_h_min.value = (getINISetting fileIni "Zone01" "hMin") as integer)catch()
try(z01_h_max.value = (getINISetting fileIni "Zone01" "hMax") as integer)catch()
try(z01_w_target.text = (getINISetting fileIni "Zone01" "wTarget") as string)catch()
try(z01_l_target.text = (getINISetting fileIni "Zone01" "lTarget") as string)catch()
try(z01_h_target.text = (getINISetting fileIni "Zone01" "hTarget") as string)catch()
try(z01_name.text = (getINISetting fileIni "Zone01" "Name") as string)catch()
try(z02_w_min.value = (getINISetting fileIni "Zone02" "wMin") as integer)catch()
try(z02_w_max.value = (getINISetting fileIni "Zone02" "wMax") as integer)catch()
try(z02_l_min.value = (getINISetting fileIni "Zone02" "lMin") as integer)catch()
try(z02_l_max.value = (getINISetting fileIni "Zone02" "lMax") as integer)catch()
try(z02_h_min.value = (getINISetting fileIni "Zone02" "hMin") as integer)catch()
try(z02_h_max.value = (getINISetting fileIni "Zone02" "hMax") as integer)catch()
try(z02_w_target.text = (getINISetting fileIni "Zone02" "wTarget") as string)catch()
try(z02_l_target.text = (getINISetting fileIni "Zone02" "lTarget") as string)catch()
try(z02_h_target.text = (getINISetting fileIni "Zone02" "hTarget") as string)catch()
try(z02_name.text = (getINISetting fileIni "Zone02" "Name") as string)catch()

```

```

try(z03_w_min.value = (getINISetting fileIni "Zone03" "wMin") as integer)catch()
try(z03_w_max.value = (getINISetting fileIni "Zone03" "wMax") as integer)catch()
try(z03_l_min.value = (getINISetting fileIni "Zone03" "lMin") as integer)catch()
try(z03_l_max.value = (getINISetting fileIni "Zone03" "lMax") as integer)catch()
try(z03_h_min.value = (getINISetting fileIni "Zone03" "hMin") as integer)catch()
try(z03_h_max.value = (getINISetting fileIni "Zone03" "hMax") as integer)catch()
try(z03_w_target.text = (getINISetting fileIni "Zone03" "wTarget") as string)catch()
try(z03_l_target.text = (getINISetting fileIni "Zone03" "lTarget") as string)catch()
try(z03_h_target.text = (getINISetting fileIni "Zone03" "hTarget") as string)catch()
try(z03_name.text = (getINISetting fileIni "Zone03" "Name") as string)catch()
try(z04_w_min.value = (getINISetting fileIni "Zone04" "wMin") as integer)catch()
try(z04_w_max.value = (getINISetting fileIni "Zone04" "wMax") as integer)catch()
try(z04_l_min.value = (getINISetting fileIni "Zone04" "lMin") as integer)catch()
try(z04_l_max.value = (getINISetting fileIni "Zone04" "lMax") as integer)catch()
try(z04_h_min.value = (getINISetting fileIni "Zone04" "hMin") as integer)catch()
try(z04_h_max.value = (getINISetting fileIni "Zone04" "hMax") as integer)catch()
try(z04_w_target.text = (getINISetting fileIni "Zone04" "wTarget") as string)catch()
try(z04_l_target.text = (getINISetting fileIni "Zone04" "lTarget") as string)catch()
try(z04_h_target.text = (getINISetting fileIni "Zone04" "hTarget") as string)catch()
try(z04_name.text = (getINISetting fileIni "Zone04" "Name") as string)catch()
try(z05_w_min.value = (getINISetting fileIni "Zone05" "wMin") as integer)catch()
try(z05_w_max.value = (getINISetting fileIni "Zone05" "wMax") as integer)catch()
try(z05_l_min.value = (getINISetting fileIni "Zone05" "lMin") as integer)catch()
try(z05_l_max.value = (getINISetting fileIni "Zone05" "lMax") as integer)catch()
try(z05_h_min.value = (getINISetting fileIni "Zone05" "hMin") as integer)catch()
try(z05_h_max.value = (getINISetting fileIni "Zone05" "hMax") as integer)catch()
try(z05_w_target.text = (getINISetting fileIni "Zone05" "wTarget") as string)catch()
try(z05_l_target.text = (getINISetting fileIni "Zone05" "lTarget") as string)catch()
try(z05_h_target.text = (getINISetting fileIni "Zone05" "hTarget") as string)catch()
try(z05_name.text = (getINISetting fileIni "Zone05" "Name") as string)catch()
)--end load dimensions settings

```

```
)--end rollout
```

```

rollout ConGen_pos "Positions" width:350 height:800
(
label z02_x_label "pos. X:" pos:[26,168] width:40 height:14
groupBox z02_pos_grp "Object02" pos:[7,145] width:330 height:120
spinner z02_x_min "min." pos:[83,168] width:60 height:16 range:[-9999,9999,0]
spinner z02_x_max "max." pos:[168,167] width:60 height:16 range:[-9999,9999,0]
label z02_y_label "pos. Y:" pos:[26,192] width:40 height:14
spinner z02_y_min "min." pos:[83,191] width:60 height:16 range:[-9999,9999,0]
spinner z02_y_max "max." pos:[168,191] width:60 height:16 range:[-9999,9999,0]
label z02_z_label "pos. Z:" pos:[26,216] width:35 height:17
spinner z02_z_min "min." pos:[83,216] width:60 height:16 range:[-9999,9999,0]
spinner z02_z_max "max." pos:[168,215] width:60 height:16 range:[-9999,9999,0]
label z03_x_label "pos. X:" pos:[26,291] width:38 height:14
groupBox z03_pos_grp "Object03" pos:[7,270] width:330 height:120
spinner z03_x_min "min." pos:[83,291] width:60 height:16 range:[-9999,9999,0]
spinner z03_x_max "max." pos:[168,291] width:60 height:16 range:[-9999,9999,0]
label z03_l_label "pos. Y:" pos:[26,317] width:40 height:14

```

spinner z03_y_min "min." pos:[83,315] width:60 height:16 range:[-9999,9999,0]
 spinner z03_y_max "max." pos:[168,315] width:60 height:16 range:[-9999,9999,0]
 label z03_z_label "pos. Z:" pos:[26,339] width:37 height:17
 spinner z03_z_min "min." pos:[83,339] width:60 height:16 range:[-9999,9999,0]
 spinner z03_z_max "max." pos:[168,339] width:60 height:16 range:[-9999,9999,0]
 label z04_x_label "pos. X:" pos:[26,419] width:42 height:14
 groupBox z04_pos_grp "Object04" pos:[8,397] width:330 height:122
 spinner z04_x_min "min." pos:[83,419] width:60 height:16 range:[-9999,9999,0]
 spinner z04_x_max "max." pos:[168,419] width:60 height:16 range:[-9999,9999,0]
 label z04_y_label "pos. Y:" pos:[26,442] width:40 height:14
 spinner z04_y_min "min." pos:[83,443] width:60 height:16 range:[-9999,9999,0]
 spinner z04_y_max "max." pos:[168,442] width:60 height:16 range:[-9999,9999,0]
 label z04_z_label "pos. Z:" pos:[26,466] width:40 height:17
 spinner z04_z_min "min." pos:[83,467] width:60 height:16 range:[-9999,9999,0]
 spinner z04_z_max "max." pos:[168,466] width:60 height:16 range:[-9999,9999,0]
 label z05_x_label "pos. X:" pos:[26,547] width:36 height:14
 groupBox z05_pos_grp "Object05" pos:[8,524] width:330 height:120
 spinner z05_x_min "min." pos:[83,547] width:60 height:16 range:[-9999,9999,0]
 spinner z05_x_max "max." pos:[168,547] width:60 height:16 range:[-9999,9999,0]
 label z05_y_label "pos. Y:" pos:[26,570] width:36 height:14
 spinner z05_y_min "min." pos:[83,570] width:60 height:16 range:[-9999,9999,0]
 spinner z05_y_max "max." pos:[168,570] width:60 height:16 range:[-9999,9999,0]
 label z05_z_label "pos. Z:" pos:[26,591] width:36 height:17
 spinner z05_z_min "min." pos:[83,592] width:60 height:16 range:[-9999,9999,0]
 spinner z05_z_max "max." pos:[168,592] width:60 height:16 range:[-9999,9999,0]
 label z01_x_label "pos. X:" pos:[26,39] width:45 height:14
 groupBox z01_pos_grp "Object01" pos:[7,17] width:330 height:120
 spinner z01_x_min "min." pos:[83,38] width:60 height:16 range:[-9999,9999,0]
 spinner z01_x_max "max." pos:[168,39] width:60 height:16 range:[-9999,9999,0]
 label z01_y_label "pos. Y:" pos:[26,63] width:39 height:14
 spinner z01_y_min "min." pos:[83,64] width:60 height:16 range:[-9999,9999,0]
 spinner z01_y_max "max." pos:[168,63] width:60 height:16 range:[-9999,9999,0]
 label z01_z_label "pos. Z:" pos:[26,89] width:38 height:17
 spinner z01_z_min "min." pos:[83,89] width:60 height:16 range:[-9999,9999,0]
 spinner z01_z_max "max." pos:[168,88] width:60 height:16 range:[-9999,9999,0]
 editText z01_x_target "target" pos:[241,39] width:75 height:16
 editText z01_y_target "target" pos:[241,64] width:75 height:16
 editText z01_z_target "target" pos:[241,87] width:75 height:16
 editText z02_x_target "target" pos:[241,167] width:75 height:16
 editText z02_y_target "target" pos:[241,191] width:75 height:16
 editText z02_z_target "target" pos:[241,215] width:75 height:16
 editText z03_x_target "target" pos:[241,291] width:75 height:16
 editText z03_y_target "target" pos:[241,315] width:75 height:16
 editText z03_z_target "target" pos:[241,339] width:75 height:16
 editText z04_x_target "target" pos:[241,418] width:75 height:16
 editText z04_y_target "target" pos:[241,443] width:75 height:16
 editText z04_z_target "target" pos:[241,467] width:75 height:16
 editText z05_x_target "target" pos:[241,547] width:75 height:16
 editText z05_y_target "target" pos:[241,570] width:75 height:16
 editText z05_z_target "target" pos:[241,592] width:75 height:16
 label z01_rotZ_label "rot. Z:" pos:[26,113] width:38 height:17

```

spinner z01_rotZ_min "min." pos:[83,112] width:60 height:16 range:[-9999,9999,0]
spinner z01_rotZ_max "max." pos:[168,111] width:60 height:16 range:[-9999,9999,0]
editText z01_rotZ_target "target" pos:[241,110] width:75 height:16
label z02_rotZ_label "rot. Z:" pos:[26,241] width:38 height:17
spinner z02_rotZ_min "min." pos:[83,240] width:60 height:16 range:[-9999,9999,0]
spinner z02_rotZ_max "max." pos:[168,239] width:60 height:16 range:[-9999,9999,0]
editText z02_rotZ_target "target" pos:[241,238] width:75 height:16
label z03_rotZ_label "rot. Z:" pos:[26,363] width:38 height:17
spinner z03_rotZ_min "min." pos:[83,363] width:60 height:16 range:[-9999,9999,0]
spinner z03_rotZ_max "max." pos:[168,362] width:60 height:16 range:[-9999,9999,0]
editText z03_rotZ_target "target" pos:[241,361] width:75 height:16
label z04_rotZ_label "rot. Z:" pos:[26,488] width:40 height:17
spinner z04_rotZ_min "min." pos:[83,489] width:60 height:16 range:[-9999,9999,0]
spinner z04_rotZ_max "max." pos:[168,489] width:60 height:16 range:[-9999,9999,0]
editText z04_rotZ_target "target" pos:[241,489] width:75 height:16
label z05_rotZ_label "rot. Z:" pos:[26,613] width:32 height:17
spinner z05_rotZ_min "min." pos:[83,614] width:60 height:16 range:[-9999,9999,0]
spinner z05_rotZ_max "max." pos:[168,614] width:60 height:16 range:[-9999,9999,0]
editText z05_rotZ_target "target" pos:[241,614] width:75 height:16
button pos_generate "Generate!!!" pos:[7,664] width:93 height:19
button pos_save "Save Settings" pos:[258,664] width:80 height:19 enabled:true
button pos_load "Load Settings" pos:[170,664] width:80 height:19 enabled:true
button z01_x_target_b ">" pos:[319,39] width:13 height:16
button z01_y_target_b ">" pos:[319,65] width:13 height:16
button z01_z_target_b ">" pos:[319,87] width:13 height:16
button z01_rotZ_target_b ">" pos:[318,110] width:13 height:16
button z02_x_target_b ">" pos:[320,167] width:13 height:16
button z02_y_target_b ">" pos:[320,192] width:13 height:16
button z02_z_target_b ">" pos:[320,215] width:13 height:16
button z02_rotZ_target_b ">" pos:[319,238] width:13 height:16
button z03_x_target_b ">" pos:[320,291] width:13 height:16
button z03_y_target_b ">" pos:[320,316] width:13 height:16
button z03_z_target_b ">" pos:[320,339] width:13 height:16
button z03_rotZ_target_b ">" pos:[319,361] width:13 height:16
button z04_x_target_b ">" pos:[320,417] width:13 height:16
button z04_y_target_b ">" pos:[320,444] width:13 height:16
button z04_z_target_b ">" pos:[321,466] width:13 height:16
button z04_rotZ_target_b ">" pos:[320,489] width:13 height:16
button z05_x_target_b ">" pos:[320,548] width:13 height:16
button z05_y_target_b ">" pos:[320,570] width:13 height:16
button z05_z_target_b ">" pos:[320,592] width:13 height:16
button z05_rotZ_target_b ">" pos:[320,614] width:13 height:16

```

```
--WORKS AFTER GENERATE PRESSED
```

```

on pos_generate pressed do(
zone01.pos.x=random z01_x_min.value z01_x_max.value
zone01.pos.y=random z01_y_min.value z01_y_max.value
zone01.pos.z=random z01_z_min.value z01_z_max.value
zone01.rotation.z_rotation=random z01_rotZ_min.value z01_rotZ_max.value

```

```
zone02.pos.x=random z02_x_min.value z02_x_max.value
```

```

zone02.pos.y=random z02_y_min.value z02_y_max.value
zone02.pos.z=random z02_z_min.value z02_z_max.value
zone02.rotation.z_rotation=random z02_rotZ_min.value z02_rotZ_max.value

```

```

zone03.pos.x=random z03_x_min.value z03_x_max.value
zone03.pos.y=random z03_y_min.value z03_y_max.value
zone03.pos.z=random z03_z_min.value z03_z_max.value
zone03.rotation.z_rotation=random z03_rotZ_min.value z03_rotZ_max.value

```

```

zone04.pos.x=random z04_x_min.value z04_x_max.value
zone04.pos.y=random z04_y_min.value z04_y_max.value
zone04.pos.z=random z04_z_min.value z04_z_max.value
zone04.rotation.z_rotation=random z04_rotZ_min.value z04_rotZ_max.value

```

```

zone05.pos.x=random z05_x_min.value z05_x_max.value
zone05.pos.y=random z05_y_min.value z05_y_max.value
zone05.pos.z=random z05_z_min.value z05_z_max.value
zone05.rotation.z_rotation=random z05_rotZ_min.value z05_rotZ_max.value

```

```

try(if z01_x_target.text!="" do zone01.pos.x=execute z01_x_target.text)catch()
try(if z01_y_target.text!="" do zone01.pos.y=execute z01_y_target.text)catch()
try(if z01_z_target.text!="" do zone01.pos.z=execute z01_z_target.text)catch()
try(if z01_rotZ_target.text!="" do zone01.rotation.z_rotation=execute
    z01_rotZ_target.text)catch()

```

```

try(if z02_x_target.text!="" do zone02.pos.x=execute z02_x_target.text)catch()
try(if z02_y_target.text!="" do zone02.pos.y=execute z02_y_target.text)catch()
try(if z02_z_target.text!="" do zone02.pos.z=execute z02_z_target.text)catch()
try(if z02_rotZ_target.text!="" do zone02.rotation.z_rotation=execute
    z02_rotZ_target.text)catch()

```

```

try(if z03_x_target.text!="" do zone03.pos.x=execute z03_x_target.text)catch()
try(if z03_y_target.text!="" do zone03.pos.y=execute z03_y_target.text)catch()
try(if z03_z_target.text!="" do zone03.pos.z=execute z03_z_target.text)catch()
try(if z03_rotZ_target.text!="" do zone03.rotation.z_rotation=execute
    z03_rotZ_target.text)catch()

```

```

try(if z04_x_target.text!="" do zone04.pos.x=execute z04_x_target.text)catch()
try(if z04_y_target.text!="" do zone04.pos.y=execute z04_y_target.text)catch()
try(if z04_z_target.text!="" do zone04.pos.z=execute z04_z_target.text)catch()
try(if z04_rotZ_target.text!="" do zone04.rotation.z_rotation=execute
    z04_rotZ_target.text)catch()

```

```

try(if z05_x_target.text!="" do zone05.pos.x=execute z05_x_target.text)catch()
try(if z05_y_target.text!="" do zone05.pos.y=execute z05_y_target.text)catch()
try(if z05_z_target.text!="" do zone05.pos.z=execute z05_z_target.text)catch()
try(if z05_rotZ_target.text!="" do zone05.rotation.z_rotation=execute
    z05_rotZ_target.text)catch()

```

```

if zone01.name!="" then z01_pos_grp.caption=zone01.name
if zone02.name!="" then z02_pos_grp.caption=zone02.name

```

```

if zone03.name!="" then z03_pos_grp.caption=zone03.name
if zone04.name!="" then z04_pos_grp.caption=zone04.name
if zone05.name!="" then z05_pos_grp.caption=zone05.name

```

```
)--END ON GENERATE
```

```
--WORKS FOR TARGET BOXES
```

```

on z01_x_target_b pressed do(
rollout tText "Target Xposition Object01" width:400 height:300(
editText targetText "Script:" pos:[5,5] width:350 height:200
on tText open do(if z01_x_target.text!="" do targetText.text=z01_x_target.text)
on tText close do(if targetText.text!="" do z01_x_target.text=targetText.text)
)
createDialog tText 380 220 true
)

```

```

on z01_y_target_b pressed do(
rollout tText "Target Yposition Object01" width:400 height:300(
editText targetText "Script:" pos:[5,5] width:350 height:200
on tText open do(if z01_y_target.text!="" do targetText.text=z01_y_target.text)
on tText close do(if targetText.text!="" do z01_y_target.text=targetText.text)
)
createDialog tText 380 220 true
)

```

```

on z01_z_target_b pressed do(
rollout tText "Target Zposition Object01" width:400 height:300(
editText targetText "Script:" pos:[5,5] width:350 height:200
on tText open do(if z01_z_target.text!="" do targetText.text=z01_z_target.text)
on tText close do(if targetText.text!="" do z01_z_target.text=targetText.text)
)
createDialog tText 380 220 true
)

```

```

on z01_rotZ_target_b pressed do(
rollout tText "Target Zrotation Object01" width:400 height:300(
editText targetText "Script:" pos:[5,5] width:350 height:200
on tText open do(if z01_rotZ_target.text!="" do targetText.text=z01_rotZ_target.text)
on tText close do(if targetText.text!="" do z01_rotZ_target.text=targetText.text)
)
createDialog tText 380 220 true
)

```

```

on z02_x_target_b pressed do(
rollout tText "Target Xposition Object02" width:400 height:300(
editText targetText "Script:" pos:[5,5] width:350 height:200
on tText open do(if z02_x_target.text!="" do targetText.text=z02_x_target.text)
on tText close do(if targetText.text!="" do z02_x_target.text=targetText.text)
)
createDialog tText 380 220 true
)

```

```

)

on z02_y_target_b pressed do(
  rollout tText "Target Yposition Object02" width:400 height:300(
    editText targetText "Script:" pos:[5,5] width:350 height:200
    on tText open do(if z02_y_target.text!=" do targetText.text=z02_y_target.text)
    on tText close do(if targetText.text!=" do z02_y_target.text=targetText.text)
  )
  createDialog tText 380 220 true
)

on z02_z_target_b pressed do(
  rollout tText "Target Zposition Object02" width:400 height:300(
    editText targetText "Script:" pos:[5,5] width:350 height:200
    on tText open do(if z02_z_target.text!=" do targetText.text=z02_z_target.text)
    on tText close do(if targetText.text!=" do z02_z_target.text=targetText.text)
  )
  createDialog tText 380 220 true
)

on z02_rotZ_target_b pressed do(
  rollout tText "Target Zrotation Object02" width:400 height:300(
    editText targetText "Script:" pos:[5,5] width:350 height:200
    on tText open do(if z02_rotZ_target.text!=" do targetText.text=z02_rotZ_target.text)
    on tText close do(if targetText.text!=" do z02_rotZ_target.text=targetText.text)
  )
  createDialog tText 380 220 true
)

on z03_x_target_b pressed do(
  rollout tText "Target Xposition Object03" width:400 height:300(
    editText targetText "Script:" pos:[5,5] width:350 height:200
    on tText open do(if z03_x_target.text!=" do targetText.text=z03_x_target.text)
    on tText close do(if targetText.text!=" do z03_x_target.text=targetText.text)
  )
  createDialog tText 380 220 true
)

on z03_y_target_b pressed do(
  rollout tText "Target Yposition Object03" width:400 height:300(
    editText targetText "Script:" pos:[5,5] width:350 height:200
    on tText open do(if z03_y_target.text!=" do targetText.text=z03_y_target.text)
    on tText close do(if targetText.text!=" do z03_y_target.text=targetText.text)
  )
  createDialog tText 380 220 true
)

on z03_z_target_b pressed do(
  rollout tText "Target Zposition Object03" width:400 height:300(
    editText targetText "Script:" pos:[5,5] width:350 height:200
    on tText open do(if z03_z_target.text!=" do targetText.text=z03_z_target.text)

```

```

on tText close do(if targetText.text!="" do z03_z_target.text=targetText.text)
)
createDialog tText 380 220 true
)

on z03_rotZ_target_b pressed do(
rollout tText "Target Zrotation Object03" width:400 height:300(
editText targetText "Script:" pos:[5,5] width:350 height:200
on tText open do(if z03_rotZ_target.text!="" do targetText.text=z03_rotZ_target.text)
on tText close do(if targetText.text!="" do z03_rotZ_target.text=targetText.text)
)
createDialog tText 380 220 true
)

on z04_x_target_b pressed do(
rollout tText "Target Xposition Object04" width:400 height:300(
editText targetText "Script:" pos:[5,5] width:350 height:200
on tText open do(if z04_x_target.text!="" do targetText.text=z04_x_target.text)
on tText close do(if targetText.text!="" do z04_x_target.text=targetText.text)
)
createDialog tText 380 220 true
)

on z04_y_target_b pressed do(
rollout tText "Target Yposition Object04" width:400 height:300(
editText targetText "Script:" pos:[5,5] width:350 height:200
on tText open do(if z04_y_target.text!="" do targetText.text=z04_y_target.text)
on tText close do(if targetText.text!="" do z04_y_target.text=targetText.text)
)
createDialog tText 380 220 true
)

on z04_z_target_b pressed do(
rollout tText "Target Zposition Object01" width:400 height:300(
editText targetText "Script:" pos:[5,5] width:350 height:200
on tText open do(if z04_z_target.text!="" do targetText.text=z04_z_target.text)
on tText close do(if targetText.text!="" do z04_z_target.text=targetText.text)
)
createDialog tText 380 220 true
)

on z04_rotZ_target_b pressed do(
rollout tText "Target Zrotation Object04" width:400 height:300(
editText targetText "Script:" pos:[5,5] width:350 height:200
on tText open do(if z04_rotZ_target.text!="" do targetText.text=z04_rotZ_target.text)
on tText close do(if targetText.text!="" do z04_rotZ_target.text=targetText.text)
)
createDialog tText 380 220 true
)

on z05_x_target_b pressed do(

```

```

rollout tText "Target Xposition Object05" width:400 height:300(
editText targetText "Script:" pos:[5,5] width:350 height:200
on tText open do(if z05_x_target.text!="" do targetText.text=z05_x_target.text)
on tText close do(if targetText.text!="" do z05_x_target.text=targetText.text)
)
createDialog tText 380 220 true
)

on z05_y_target_b pressed do(
rollout tText "Target Yposition Object05" width:400 height:300(
editText targetText "Script:" pos:[5,5] width:350 height:200
on tText open do(if z05_y_target.text!="" do targetText.text=z05_y_target.text)
on tText close do(if targetText.text!="" do z05_y_target.text=targetText.text)
)
createDialog tText 380 220 true
)

on z05_z_target_b pressed do(
rollout tText "Target Zposition Object01" width:400 height:300(
editText targetText "Script:" pos:[5,5] width:350 height:200
on tText open do(if z05_z_target.text!="" do targetText.text=z05_z_target.text)
on tText close do(if targetText.text!="" do z05_z_target.text=targetText.text)
)
createDialog tText 380 220 true
)

on z05_rotZ_target_b pressed do(
rollout tText "Target Zrotation Object05" width:400 height:300(
editText targetText "Script:" pos:[5,5] width:350 height:200
on tText open do(if z05_rotZ_target.text!="" do targetText.text=z05_rotZ_target.text)
on tText close do(if targetText.text!="" do z05_rotZ_target.text=targetText.text)
)
createDialog tText 380 220 true
)

--END TARGET BOXES SETTINGS

on pos_save pressed do(

fileIni=getSaveFileName caption:"Save as"
setINISetting fileIni "Zone01" "xMin" (z01_x_min.value as string)
setINISetting fileIni "Zone01" "xMax" (z01_x_max.value as string)
setINISetting fileIni "Zone01" "yMin" (z01_y_min.value as string)
setINISetting fileIni "Zone01" "yMax" (z01_y_max.value as string)
setINISetting fileIni "Zone01" "zMin" (z01_z_min.value as string)
setINISetting fileIni "Zone01" "zMax" (z01_z_max.value as string)
setINISetting fileIni "Zone01" "rMin" (z01_rotZ_min.value as string)
setINISetting fileIni "Zone01" "rMax" (z01_rotZ_max.value as string)
setINISetting fileIni "Zone01" "xTarget" (z01_x_target.text as string)
setINISetting fileIni "Zone01" "yTarget" (z01_y_target.text as string)
setINISetting fileIni "Zone01" "zTarget" (z01_z_target.text as string)

```

```
setINISetting fileIni "Zone01" "rTarget" (z01_rotZ_target.text as string)
setINISetting fileIni "Zone02" "xMin" (z02_x_min.value as string)
setINISetting fileIni "Zone02" "xMax" (z02_x_max.value as string)
setINISetting fileIni "Zone02" "yMin" (z02_y_min.value as string)
setINISetting fileIni "Zone02" "yMax" (z02_y_max.value as string)
setINISetting fileIni "Zone02" "zMin" (z02_z_min.value as string)
setINISetting fileIni "Zone02" "zMax" (z02_z_max.value as string)
setINISetting fileIni "Zone02" "rMin" (z02_rotZ_min.value as string)
setINISetting fileIni "Zone02" "rMax" (z02_rotZ_max.value as string)
setINISetting fileIni "Zone02" "xTarget" (z02_x_target.text as string)
setINISetting fileIni "Zone02" "yTarget" (z02_y_target.text as string)
setINISetting fileIni "Zone02" "zTarget" (z02_z_target.text as string)
setINISetting fileIni "Zone02" "rTarget" (z02_rotZ_target.text as string)
setINISetting fileIni "Zone03" "xMin" (z03_x_min.value as string)
setINISetting fileIni "Zone03" "xMax" (z03_x_max.value as string)
setINISetting fileIni "Zone03" "yMin" (z03_y_min.value as string)
setINISetting fileIni "Zone03" "yMax" (z03_y_max.value as string)
setINISetting fileIni "Zone03" "zMin" (z03_z_min.value as string)
setINISetting fileIni "Zone03" "zMax" (z03_z_max.value as string)
setINISetting fileIni "Zone03" "rMin" (z03_rotZ_min.value as string)
setINISetting fileIni "Zone03" "rMax" (z03_rotZ_max.value as string)
setINISetting fileIni "Zone03" "xTarget" (z03_x_target.text as string)
setINISetting fileIni "Zone03" "yTarget" (z03_y_target.text as string)
setINISetting fileIni "Zone03" "zTarget" (z03_z_target.text as string)
setINISetting fileIni "Zone03" "rTarget" (z03_rotZ_target.text as string)
setINISetting fileIni "Zone04" "xMin" (z04_x_min.value as string)
setINISetting fileIni "Zone04" "xMax" (z04_x_max.value as string)
setINISetting fileIni "Zone04" "yMin" (z04_y_min.value as string)
setINISetting fileIni "Zone04" "yMax" (z04_y_max.value as string)
setINISetting fileIni "Zone04" "zMin" (z04_z_min.value as string)
setINISetting fileIni "Zone04" "zMax" (z04_z_max.value as string)
setINISetting fileIni "Zone04" "rMin" (z04_rotZ_min.value as string)
setINISetting fileIni "Zone04" "rMax" (z04_rotZ_max.value as string)
setINISetting fileIni "Zone04" "xTarget" (z04_x_target.text as string)
setINISetting fileIni "Zone04" "yTarget" (z04_y_target.text as string)
setINISetting fileIni "Zone04" "zTarget" (z04_z_target.text as string)
setINISetting fileIni "Zone04" "rTarget" (z04_rotZ_target.text as string)
setINISetting fileIni "Zone05" "xMin" (z05_x_min.value as string)
setINISetting fileIni "Zone05" "xMax" (z05_x_max.value as string)
setINISetting fileIni "Zone05" "yMin" (z05_y_min.value as string)
setINISetting fileIni "Zone05" "yMax" (z05_y_max.value as string)
setINISetting fileIni "Zone05" "zMin" (z05_z_min.value as string)
setINISetting fileIni "Zone05" "zMax" (z05_z_max.value as string)
setINISetting fileIni "Zone05" "rMin" (z05_rotZ_min.value as string)
setINISetting fileIni "Zone05" "rMax" (z05_rotZ_max.value as string)
setINISetting fileIni "Zone05" "xTarget" (z05_x_target.text as string)
setINISetting fileIni "Zone05" "yTarget" (z05_y_target.text as string)
setINISetting fileIni "Zone05" "zTarget" (z05_z_target.text as string)
setINISetting fileIni "Zone05" "rTarget" (z05_rotZ_target.text as string)
)
```

```
on pos_load pressed do(
fileIni=getOpenFileName caption:"Load"
```

```
try(z01_x_min.value = (getINISetting fileIni "Zone01" "xMin") as integer)catch()
try(z01_x_max.value = (getINISetting fileIni "Zone01" "xMax") as integer)catch()
try(z01_y_min.value = (getINISetting fileIni "Zone01" "yMin") as integer)catch()
try(z01_y_max.value = (getINISetting fileIni "Zone01" "yMax") as integer)catch()
try(z01_z_min.value = (getINISetting fileIni "Zone01" "zMin") as integer)catch()
try(z01_z_max.value = (getINISetting fileIni "Zone01" "zMax") as integer)catch()
try(z01_rotZ_min.value = (getINISetting fileIni "Zone01" "rMin") as integer)catch()
try(z01_rotZ_max.value = (getINISetting fileIni "Zone01" "rMax") as integer)catch()
try(z01_x_target.text = (getINISetting fileIni "Zone01" "xTarget") as string)catch()
try(z01_y_target.text = (getINISetting fileIni "Zone01" "yTarget") as string)catch()
try(z01_z_target.text = (getINISetting fileIni "Zone01" "zTarget") as string)catch()
try(z01_rotZ_target.text = (getINISetting fileIni "Zone01" "rTarget") as string)catch()
```

```
try(z02_x_min.value = (getINISetting fileIni "Zone02" "xMin") as integer)catch()
try(z02_x_max.value = (getINISetting fileIni "Zone02" "xMax") as integer)catch()
try(z02_y_min.value = (getINISetting fileIni "Zone02" "yMin") as integer)catch()
try(z02_y_max.value = (getINISetting fileIni "Zone02" "yMax") as integer)catch()
try(z02_z_min.value = (getINISetting fileIni "Zone02" "zMin") as integer)catch()
try(z02_z_max.value = (getINISetting fileIni "Zone02" "zMax") as integer)catch()
try(z02_rotZ_min.value = (getINISetting fileIni "Zone02" "rMin") as integer)catch()
try(z02_rotZ_max.value = (getINISetting fileIni "Zone02" "rMax") as integer)catch()
try(z02_x_target.text = (getINISetting fileIni "Zone02" "xTarget") as string)catch()
try(z02_y_target.text = (getINISetting fileIni "Zone02" "yTarget") as string)catch()
try(z02_z_target.text = (getINISetting fileIni "Zone02" "zTarget") as string)catch()
try(z02_rotZ_target.text = (getINISetting fileIni "Zone02" "rTarget") as string)catch()
```

```
try(z03_x_min.value = (getINISetting fileIni "Zone03" "xMin") as integer)catch()
try(z03_x_max.value = (getINISetting fileIni "Zone03" "xMax") as integer)catch()
try(z03_y_min.value = (getINISetting fileIni "Zone03" "yMin") as integer)catch()
try(z03_y_max.value = (getINISetting fileIni "Zone03" "yMax") as integer)catch()
try(z03_z_min.value = (getINISetting fileIni "Zone03" "zMin") as integer)catch()
try(z03_z_max.value = (getINISetting fileIni "Zone03" "zMax") as integer)catch()
try(z03_rotZ_min.value = (getINISetting fileIni "Zone03" "rMin") as integer)catch()
try(z03_rotZ_max.value = (getINISetting fileIni "Zone03" "rMax") as integer)catch()
try(z03_x_target.text = (getINISetting fileIni "Zone03" "xTarget") as string)catch()
try(z03_y_target.text = (getINISetting fileIni "Zone03" "yTarget") as string)catch()
try(z03_z_target.text = (getINISetting fileIni "Zone03" "zTarget") as string)catch()
try(z03_rotZ_target.text = (getINISetting fileIni "Zone03" "rTarget") as string)catch()
```

```
try(z04_x_min.value = (getINISetting fileIni "Zone04" "xMin") as integer)catch()
try(z04_x_max.value = (getINISetting fileIni "Zone04" "xMax") as integer)catch()
try(z04_y_min.value = (getINISetting fileIni "Zone04" "yMin") as integer)catch()
try(z04_y_max.value = (getINISetting fileIni "Zone04" "yMax") as integer)catch()
try(z04_z_min.value = (getINISetting fileIni "Zone04" "zMin") as integer)catch()
try(z04_z_max.value = (getINISetting fileIni "Zone04" "zMax") as integer)catch()
try(z04_rotZ_min.value = (getINISetting fileIni "Zone04" "rMin") as integer)catch()
try(z04_rotZ_max.value = (getINISetting fileIni "Zone04" "rMax") as integer)catch()
try(z04_x_target.text = (getINISetting fileIni "Zone04" "xTarget") as string)catch()
```

```
try(z04_y_target.text = (getINISetting fileIni "Zone04" "yTarget") as string)catch()
try(z04_z_target.text = (getINISetting fileIni "Zone04" "zTarget") as string)catch()
try(z04_rotZ_target.text = (getINISetting fileIni "Zone04" "rTarget") as string)catch()
```

```
try(z05_x_min.value = (getINISetting fileIni "Zone05" "xMin") as integer)catch()
try(z05_x_max.value = (getINISetting fileIni "Zone05" "xMax") as integer)catch()
try(z05_y_min.value = (getINISetting fileIni "Zone05" "yMin") as integer)catch()
try(z05_y_max.value = (getINISetting fileIni "Zone05" "yMax") as integer)catch()
try(z05_z_min.value = (getINISetting fileIni "Zone05" "zMin") as integer)catch()
try(z05_z_max.value = (getINISetting fileIni "Zone05" "zMax") as integer)catch()
try(z05_rotZ_min.value = (getINISetting fileIni "Zone05" "rMin") as integer)catch()
try(z05_rotZ_max.value = (getINISetting fileIni "Zone05" "rMax") as integer)catch()
try(z05_x_target.text = (getINISetting fileIni "Zone05" "xTarget") as string)catch()
try(z05_y_target.text = (getINISetting fileIni "Zone05" "yTarget") as string)catch()
try(z05_z_target.text = (getINISetting fileIni "Zone05" "zTarget") as string)catch()
try(z05_rotZ_target.text = (getINISetting fileIni "Zone05" "rTarget") as string)catch()
)
)
```

```
rollout ConGen_mod "Modifiers" width:350 height:700(
```

```
label bend_angle_label "angle:" pos:[26,168] width:40 height:14
groupBox bend_grp "Bend" pos:[6,143] width:330 height:150
spinner bend_min "min." pos:[83,168] width:60 height:16 range:[-9999,9999,0]
spinner bend_max "max." pos:[168,167] width:60 height:16 range:[-9999,9999,0]
label bend_direc "direction:" pos:[26,192] width:47 height:14
spinner bend_direc_min "min." pos:[83,191] width:60 height:16 range:[-9999,9999,0]
spinner bend_direc_max "max." pos:[168,191] width:60 height:16 range:[-9999,9999,0]
label spher_percent_label "percent:" pos:[26,39] width:45 height:14
groupBox spher_grp "Spherify" pos:[7,17] width:330 height:120
spinner spher_min "min." pos:[83,39] width:60 height:16 range:[-9999,9999,0]
spinner spher_max "max." pos:[168,39] width:60 height:16 range:[-9999,9999,0]
checkbox sphere_zone01 "Object01" pos:[263,36] width:61 height:20 checked:false
checkbox sphere_zone02 "Object02" pos:[263,54] width:61 height:20 checked:false
checkbox sphere_zone03 "Object03" pos:[263,72] width:61 height:20 checked:false
checkbox sphere_zone04 "Object04" pos:[263,90] width:61 height:20 checked:false
checkbox sphere_zone05 "Object05" pos:[263,108] width:61 height:20 checked:false
checkbox bend_zone01 "Object01" pos:[262,163] width:61 height:20 checked:false
checkbox bend_zone02 "Object02" pos:[262,181] width:61 height:20 checked:false
checkbox bend_zone03 "Object03" pos:[262,199] width:61 height:20 checked:false
checkbox bend_zone04 "Object04" pos:[262,217] width:61 height:20 checked:false
checkbox bend_zone05 "Object05" pos:[262,235] width:61 height:20 checked:false
label skew_amount "amount:" pos:[25,325] width:40 height:14
groupBox skew_grp "Skew" pos:[6,300] width:330 height:161
spinner skew_amo_min "min." pos:[82,325] width:60 height:16 range:[-9999,9999,0]
spinner skew_amo_max "max." pos:[167,324] width:60 height:16 range:[-9999,9999,0]
label skew_direc "direction:" pos:[25,349] width:45 height:14
spinner skew_direc_min "min." pos:[83,348] width:60 height:16 range:[-9999,9999,0]
spinner skew_direc_max "max." pos:[167,348] width:60 height:16 range:[-9999,9999,0]
```

```
checkbox skew_zone01 "Object01" pos:[261,320] width:61 height:20 checked:false
checkbox skew_zone02 "Object02" pos:[261,338] width:61 height:20 checked:false
checkbox skew_zone03 "Object03" pos:[261,356] width:61 height:20 checked:false
checkbox skew_zone04 "Object04" pos:[261,374] width:61 height:20 checked:false
checkbox skew_zone05 "Object05" pos:[261,392] width:61 height:20 checked:false
checkbox skew_limit "limit effect" pos:[24,379] width:71 height:20 checked:false
label skew_upperL "upper L:" pos:[23,401] width:45 height:14
spinner skew_upperL_min "min." pos:[83,400] width:60 height:16 range:[-9999,9999,0]
spinner skew_upperL_max "max." pos:[167,400] width:60 height:16 range:[-9999,9999,0]
label skew_lowerL "lower L:" pos:[24,422] width:45 height:14
spinner skew_lowerL_min "min." pos:[83,421] width:60 height:16 range:[-9999,9999,0]
spinner skew_lowerL_max "max." pos:[167,421] width:60 height:16 range:[-9999,9999,0]
checkbox bend_limit "limit effect" pos:[24,221] width:71 height:20 checked:false
label bend_upperL "upper L:" pos:[23,243] width:45 height:14
spinner bend_upperL_min "min." pos:[83,242] width:60 height:16 range:[-9999,9999,0]
spinner bend_upperL_max "max." pos:[167,242] width:60 height:16 range:[-9999,9999,0]
label bend_lowerL "lower L:" pos:[24,264] width:45 height:14
spinner bend_lowerL_min "min." pos:[83,263] width:60 height:16 range:[-9999,9999,0]
spinner bend_lowerL_max "max." pos:[167,263] width:60 height:16 range:[-9999,9999,0]
label taper_amo "amount:" pos:[24,495] width:40 height:14
groupBox taper_grp "Taper" pos:[6,470] width:330 height:161
spinner taper_amo_min "min." pos:[82,495] width:60 height:16 range:[-10,10,0]
spinner taper_amo_max "max." pos:[167,494] width:60 height:16 range:[-10,10,0]
label taper_curve "curve:" pos:[25,519] width:45 height:14
spinner taper_curve_min "min." pos:[83,518] width:60 height:16 range:[-10,10,0]
spinner taper_curve_max "max." pos:[167,518] width:60 height:16 range:[-10,10,0]
checkbox taper_zone01 "Object01" pos:[261,490] width:61 height:20 checked:false
checkbox taper_zone02 "Object02" pos:[261,508] width:61 height:20 checked:false
checkbox taper_zone03 "Object03" pos:[261,526] width:61 height:20 checked:false
checkbox taper_zone04 "Object04" pos:[261,544] width:61 height:20 checked:false
checkbox taper_zone05 "Object05" pos:[261,562] width:61 height:20 checked:false
checkbox taper_limit "limit effect" pos:[24,549] width:71 height:20 checked:false
label taper_upperL "upper L:" pos:[23,571] width:45 height:14
spinner taper_upperL_min "min." pos:[83,570] width:60 height:16 range:[-9999,9999,0]
spinner taper_upperL_max "max." pos:[167,570] width:60 height:16 range:[-9999,9999,0]
label taper_lowerL "lower L:" pos:[24,592] width:45 height:14
spinner taper_lowerL_min "min." pos:[83,591] width:60 height:16 range:[-9999,9999,0]
spinner taper_lowerL_max "max." pos:[167,590] width:60 height:16 range:[-9999,9999,0]
button mod_generate "Generate!!!" pos:[7,664] width:93 height:19
button mod_save "Save Settings" pos:[258,664] width:80 height:19 enabled:true
button mod_load "Load Settings" pos:[170,664] width:80 height:19 enabled:true
```

```
on mod_generate pressed do(
```

```
--UPDATING ZONE NAMES
```

```
sphere_zone01.caption=zone01.name
```

```
sphere_zone02.caption=zone02.name
```

```
sphere_zone03.caption=zone03.name
```

```
sphere_zone04.caption=zone04.name
```

```
sphere_zone05.caption=zone05.name
```

```
taper_zone01.caption=zone01.name
taper_zone02.caption=zone02.name
taper_zone03.caption=zone03.name
taper_zone04.caption=zone04.name
taper_zone05.caption=zone05.name
```

```
skew_zone01.caption=zone01.name
skew_zone02.caption=zone02.name
skew_zone03.caption=zone03.name
skew_zone04.caption=zone04.name
skew_zone05.caption=zone05.name
```

```
bend_zone01.caption=zone01.name
bend_zone02.caption=zone02.name
bend_zone03.caption=zone03.name
bend_zone04.caption=zone04.name
bend_zone05.caption=zone05.name
```

--APPLYING MODIFIERS

```
if skew_zone01.checked then(
try (if zone01.modifiers[#skew]!==undefined then deleteModifier zone01
    zone01.modifiers[#skew])catch()
addmodifier zone01 (skew())
zone01.skew.axis=3
zone01.skew.amount=random skew_amo_min.value skew_amo_max.value
zone01.skew.direction=random skew_direc_min.value skew_direc_max.value
if skew_limit.checked then(
zone01.skew.limit=true
zone01.skew.upperlimit=random skew_upperL_min.value skew_upperL_max.value
zone01.skew.lowerlimit=random skew_lowerL_min.value skew_lowerL_max.value
)
)
```

```
if skew_zone02.checked then(
try (if zone02.modifiers[#skew]!==undefined then deleteModifier zone02
    zone02.modifiers[#skew])catch()
addmodifier zone02 (skew())
zone02.skew.axis=3
zone02.skew.amount=random skew_amo_min.value skew_amo_max.value
zone02.skew.direction=random skew_direc_min.value skew_direc_max.value
if skew_limit.checked then(
zone02.skew.limit=true
zone02.skew.upperlimit=random skew_upperL_min.value skew_upperL_max.value
zone02.skew.lowerlimit=random skew_lowerL_min.value skew_lowerL_max.value
)
)
```

```
if skew_zone03.checked then(
```

```

try (if zone03.modifiers[#skew]!:=undefined then deleteModifier zone03
    zone03.modifiers[#skew])catch()
addmodifier zone03 (skew())
zone03.skew.axis=3
zone03.skew.amount=random skew_amo_min.value skew_amo_max.value
zone03.skew.direction=random skew_direc_min.value skew_direc_max.value
if skew_limit.checked then(
zone03.skew.limit=true
zone03.skew.upperlimit=random skew_upperL_min.value skew_upperL_max.value
zone03.skew.lowerlimit=random skew_lowerL_min.value skew_lowerL_max.value
)
)

if skew_zone04.checked then(
try (if zone04.modifiers[#skew]!:=undefined then deleteModifier zone04
    zone04.modifiers[#skew])catch()
addmodifier zone04 (skew())
zone04.skew.direction=random skew_direc_min.value skew_direc_max.value
if skew_limit.checked then(
zone04.skew.limit=true
zone04.skew.upperlimit=random skew_upperL_min.value skew_upperL_max.value
zone04.skew.lowerlimit=random skew_lowerL_min.value skew_lowerL_max.value
)
)

if skew_zone05.checked then(
try (if zone05.modifiers[#skew]!:=undefined then deleteModifier zone05
    zone05.modifiers[#skew])catch()
addmodifier zone05 (skew())
zone05.skew.axis=3
zone05.skew.amount=random skew_amo_min.value skew_amo_max.value
zone05.skew.direction=random skew_direc_min.value skew_direc_max.value
if skew_limit.checked then(
zone05.skew.limit=true
zone05.skew.upperlimit=random skew_upperL_min.value skew_upperL_max.value
zone05.skew.lowerlimit=random skew_lowerL_min.value skew_lowerL_max.value
)
)

if sphere_zone01.checked then(
try (if zone01.modifiers[#spherify]!:=undefined then deleteModifier zone01
    zone01.modifiers[#spherify])catch()
addmodifier zone01 (spherify())
zone01.spherify.percent=random spher_min.value spher_max.value
)

if sphere_zone02.checked then(
try (if zone02.modifiers[#spherify]!:=undefined then deleteModifier zone02
    zone02.modifiers[#spherify])catch()
addmodifier zone02 (spherify())
zone02.spherify.percent=random spher_min.value spher_max.value

```

```

)

if sphere_zone03.checked then(
try (if zone03.modifiers[#spherify]!=undefined then deleteModifier zone03
    zone03.modifiers[#spherify])catch()
addmodifier zone03 (spherify())
zone03.spherify.percent=random spher_min.value spher_max.value
)

if sphere_zone04.checked then(
try (if zone04.modifiers[#spherify]!=undefined then deleteModifier zone04
    zone04.modifiers[#spherify])catch()
addmodifier zone04 (spherify())
zone04.spherify.percent=random spher_min.value spher_max.value
)

if sphere_zone05.checked then(
try (if zone05.modifiers[#spherify]!=undefined then deleteModifier zone05
    zone05.modifiers[#spherify])catch()
addmodifier zone05 (spherify())
zone05.spherify.percent=random spher_min.value spher_max.value
)

if taper_zone01.checked then(
try (if zone01.modifiers[#taper]!=undefined then deleteModifier zone01
    zone01.modifiers[#taper])catch()
addmodifier zone01 (taper())
zone01.taper.amount=random taper_amo_min.value taper_amo_max.value
zone01.taper.curve=random taper_curve_min.value taper_curve_max.value
zone01.taper.primaryaxis=2
if taper_limit.checked then(
zone01.taper.limit=true
zone01.taper.upperlimit=random taper_upperL_min.value taper_upperL_max.value
zone01.taper.lowerlimit=random taper_lowerL_min.value taper_lowerL_max.value
)
)

if taper_zone02.checked then(
try (if zone02.modifiers[#taper]!=undefined then deleteModifier zone02
    zone02.modifiers[#taper])catch()
addmodifier zone02 (taper())
zone02.taper.amount=random taper_amo_min.value taper_amo_max.value
zone02.taper.curve=random taper_curve_min.value taper_curve_max.value
zone02.taper.primaryaxis=2
if taper_limit.checked then(
zone02.taper.limit=true
zone02.taper.upperlimit=random taper_upperL_min.value taper_upperL_max.value
zone02.taper.lowerlimit=random taper_lowerL_min.value taper_lowerL_max.value
)
)
)

```

```

if taper_zone03.checked then(
try (if zone03.modifiers[#taper]!==undefined then deleteModifier zone03
    zone03.modifiers[#taper])catch()
addmodifier zone03 (taper())
zone03.taper.amount=random taper_amo_min.value taper_amo_max.value
zone03.taper.curve=random taper_curve_min.value taper_curve_max.value
zone03.taper.primaryaxis=2
if taper_limit.checked then(
zone03.taper.limit=true
zone03.taper.upperlimit=random taper_upperL_min.value taper_upperL_max.value
zone03.taper.lowerlimit=random taper_lowerL_min.value taper_lowerL_max.value
)
)

```

```

if taper_zone04.checked then(
try (if zone04.modifiers[#taper]!==undefined then deleteModifier zone04
    zone04.modifiers[#taper])catch()
addmodifier zone04 (taper())
zone04.taper.amount=random taper_amo_min.value taper_amo_max.value
zone04.taper.curve=random taper_curve_min.value taper_curve_max.value
zone04.taper.primaryaxis=2
if taper_limit.checked then(
zone04.taper.limit=true
zone04.taper.upperlimit=random taper_upperL_min.value taper_upperL_max.value
zone04.taper.lowerlimit=random taper_lowerL_min.value taper_lowerL_max.value
)
)

```

```

if taper_zone05.checked then(
try (if zone05.modifiers[#taper]!==undefined then deleteModifier zone05
    zone05.modifiers[#taper])catch()
addmodifier zone05 (taper())
zone05.taper.amount=random taper_amo_min.value taper_amo_max.value
zone05.taper.curve=random taper_curve_min.value taper_curve_max.value
zone05.taper.primaryaxis=2
if taper_limit.checked then(
zone05.taper.limit=true
zone05.taper.upperlimit=random taper_upperL_min.value taper_upperL_max.value
zone05.taper.lowerlimit=random taper_lowerL_min.value taper_lowerL_max.value
)
)

```

```

if bend_zone01.checked then(
try (if zone01.modifiers[#bend]!==undefined then deleteModifier zone01
    zone01.modifiers[#bend])catch()
addmodifier zone01 (bend())
zone01.bend.angle=random bend_min.value bend_max.value
zone01.bend.direction=random bend_dirac_min.value bend_dirac_max.value
if bend_limit.checked then(
zone01.bend.limit=true
zone01.bend.upperlimit=random bend_upperL_min.value bend_upperL_max.value

```

```

zone01.bend.lowerlimit=random bend_lowerL_min.value bend_lowerL_max.value
)
)

```

```

if bend_zone02.checked then(
try (if zone02.modifiers[#bend]!=undefined then deleteModifier zone02
    zone02.modifiers[#bend])catch()
addmodifier zone02 (bend())
zone02.bend.angle=random bend_min.value bend_max.value
zone02.bend.direction=random bend_dirc_min.value bend_dirc_max.value
if bend_limit.checked then(
zone02.bend.upperlimit=random bend_upperL_min.value bend_upperL_max.value
zone02.bend.lowerlimit=random bend_lowerL_min.value bend_lowerL_max.value
)
)

```

```

if bend_zone03.checked then(
try (if zone03.modifiers[#bend]!=undefined then deleteModifier zone03
    zone03.modifiers[#bend])catch()
addmodifier zone03 (bend())
zone03.bend.angle=random bend_min.value bend_max.value
zone03.bend.direction=random bend_dirc_min.value bend_dirc_max.value
if bend_limit.checked then(
zone03.bend.limit=true
zone03.bend.upperlimit=random bend_upperL_min.value bend_upperL_max.value
)
)

```

```

if bend_zone04.checked then(
try (if zone04.modifiers[#bend]!=undefined then deleteModifier zone04
    zone04.modifiers[#bend])catch()
addmodifier zone04 (bend())
zone04.bend.angle=random bend_min.value bend_max.value
zone04.bend.direction=random bend_dirc_min.value bend_dirc_max.value
if bend_limit.checked then(
zone04.bend.limit=true
zone04.bend.upperlimit=random bend_upperL_min.value bend_upperL_max.value
zone04.bend.lowerlimit=random bend_lowerL_min.value bend_lowerL_max.value
)
)

```

```

if bend_zone05.checked then(
try (if zone05.modifiers[#bend]!=undefined then deleteModifier zone05
    zone05.modifiers[#bend])catch()
addmodifier zone05 (bend())
zone05.bend.angle=random bend_min.value bend_max.value
zone05.bend.direction=random bend_dirc_min.value bend_dirc_max.value
if bend_limit.checked then(
zone05.bend.limit=true
zone05.bend.upperlimit=random bend_upperL_min.value bend_upperL_max.value
zone05.bend.lowerlimit=random bend_lowerL_min.value bend_lowerL_max.value
)
)

```

```
)
)
)
```

```
on mod_save pressed do(
fileIni=getSaveFileName caption:"Save as"
setINISetting fileIni "Zone01" "Spherify" (sphere_zone01.triState as string)
setINISetting fileIni "Zone01" "Skew" (skew_zone01.triState as string)
setINISetting fileIni "Zone01" "Bend" (bend_zone01.triState as string)
setINISetting fileIni "Zone01" "Taper" (taper_zone01.triState as string)

setINISetting fileIni "Zone02" "Spherify" (sphere_zone02.triState as string)
setINISetting fileIni "Zone02" "Skew" (skew_zone02.triState as string)
setINISetting fileIni "Zone02" "Bend" (bend_zone02.triState as string)
setINISetting fileIni "Zone02" "Taper" (taper_zone02.triState as string)

setINISetting fileIni "Zone03" "Spherify" (sphere_zone03.triState as string)
setINISetting fileIni "Zone03" "Skew" (skew_zone03.triState as string)
setINISetting fileIni "Zone03" "Bend" (bend_zone03.triState as string)
setINISetting fileIni "Zone03" "Taper" (taper_zone03.triState as string)

setINISetting fileIni "Zone04" "Spherify" (sphere_zone04.triState as string)
setINISetting fileIni "Zone04" "Skew" (skew_zone04.triState as string)
setINISetting fileIni "Zone04" "Bend" (bend_zone04.triState as string)
setINISetting fileIni "Zone04" "Taper" (taper_zone04.triState as string)

setINISetting fileIni "Zone05" "Spherify" (sphere_zone05.triState as string)
setINISetting fileIni "Zone05" "Skew" (skew_zone05.triState as string)
setINISetting fileIni "Zone05" "Bend" (bend_zone05.triState as string)
setINISetting fileIni "Zone05" "Taper" (taper_zone05.triState as string)

setINISetting fileIni "Spherify" "minPercent" (spher_min.value as string)
setINISetting fileIni "Spherify" "maxPercent" (spher_max.value as string)

setINISetting fileIni "Bend" "minAngle" (bend_min.value as string)
setINISetting fileIni "Bend" "maxAngle" (bend_max.value as string)
setINISetting fileIni "Bend" "minDirection" (bend_dirac_min.value as string)
setINISetting fileIni "Bend" "maxDirection" (bend_dirac_max.value as string)
setINISetting fileIni "Bend" "Limit" (bend_limit.triState as string)
setINISetting fileIni "Bend" "upperLimitMin" (bend_upperL_min.value as string)
setINISetting fileIni "Bend" "upperLimitMax" (bend_upperL_max.value as string)
setINISetting fileIni "Bend" "lowerLimitMin" (bend_lowerL_min.value as string)
setINISetting fileIni "Bend" "lowerLimitMax" (bend_lowerL_max.value as string)

setINISetting fileIni "Skew" "minAngle" (skew_amo_min.value as string)
setINISetting fileIni "Skew" "maxAngle" (skew_amo_max.value as string)
setINISetting fileIni "Skew" "minDirection" (skew_dirac_min.value as string)
setINISetting fileIni "Skew" "maxDirection" (skew_dirac_max.value as string)
setINISetting fileIni "Skew" "Limit" (skew_limit.triState as string)
setINISetting fileIni "Skew" "upperLimitMin" (skew_upperL_min.value as string)
setINISetting fileIni "Skew" "upperLimitMax" (skew_upperL_max.value as string)
```

```
setINISetting fileIni "Skew" "lowerLimitMin" (skew_lowerL_min.value as string)
setINISetting fileIni "Skew" "lowerLimitMax" (skew_lowerL_max.value as string)
```

```
setINISetting fileIni "Taper" "minAngle" (taper_amo_min.value as string)
setINISetting fileIni "Taper" "maxAngle" (taper_amo_max.value as string)
setINISetting fileIni "Taper" "minDirection" (taper_curve_min.value as string)
setINISetting fileIni "Taper" "maxDirection" (taper_curve_max.value as string)
setINISetting fileIni "Taper" "Limit" (taper_limit.triState as string)
setINISetting fileIni "Taper" "upperLimitMin" (taper_upperL_min.value as string)
setINISetting fileIni "Taper" "upperLimitMax" (taper_upperL_max.value as string)
setINISetting fileIni "Taper" "lowerLimitMin" (taper_lowerL_min.value as string)
setINISetting fileIni "Taper" "lowerLimitMax" (taper_lowerL_max.value as string)
)
```

```
on mod_load pressed do(
fileIni=getOpenFileName caption:"Load"
```

```
try(sphere_zone01.triState = (getINISetting fileIni "Zone01" "Spherify") as integer)catch()
try(skew_zone01.triState = (getINISetting fileIni "Zone01" "Skew") as integer)catch()
try(bend_zone01.triState = (getINISetting fileIni "Zone01" "Bend") as integer)catch()
try(taper_zone01.triState = (getINISetting fileIni "Zone01" "Taper") as integer)catch()
```

```
try(sphere_zone02.triState = (getINISetting fileIni "Zone02" "Spherify") as integer)catch()
try(skew_zone02.triState = (getINISetting fileIni "Zone02" "Skew") as integer)catch()
try(bend_zone02.triState = (getINISetting fileIni "Zone02" "Bend") as integer)catch()
try(taper_zone02.triState = (getINISetting fileIni "Zone02" "Taper") as integer)catch()
```

```
try(sphere_zone03.triState = (getINISetting fileIni "Zone03" "Spherify") as integer)catch()
try(skew_zone03.triState = (getINISetting fileIni "Zone03" "Skew") as integer)catch()
try(bend_zone03.triState = (getINISetting fileIni "Zone03" "Bend") as integer)catch()
try(taper_zone03.triState = (getINISetting fileIni "Zone03" "Taper") as integer)catch()
```

```
try(sphere_zone04.triState = (getINISetting fileIni "Zone04" "Spherify") as integer)catch()
try(skew_zone04.triState = (getINISetting fileIni "Zone04" "Skew") as integer)catch()
try(bend_zone04.triState = (getINISetting fileIni "Zone04" "Bend") as integer)catch()
try(taper_zone04.triState = (getINISetting fileIni "Zone04" "Taper") as integer)catch()
```

```
try(sphere_zone05.triState = (getINISetting fileIni "Zone05" "Spherify") as integer)catch()
try(skew_zone05.triState = (getINISetting fileIni "Zone05" "Skew") as integer)catch()
try(bend_zone05.triState = (getINISetting fileIni "Zone05" "Bend") as integer)catch()
try(taper_zone05.triState = (getINISetting fileIni "Zone05" "Taper") as integer)catch()
```

```
try(spher_min.value = (getINISetting fileIni "Spherify" "minPercent") as integer)catch()
try(spher_max.value = (getINISetting fileIni "Spherify" "maxPercent") as integer)catch()
```

```
try(bend_min.value = (getINISetting fileIni "Bend" "minAngle") as integer)catch()
try(bend_max.value = (getINISetting fileIni "Bend" "maxAngle") as integer)catch()
try(bend_dirac_min.value = (getINISetting fileIni "Bend" "minDirection") as integer)catch()
try(bend_dirac_max.value = (getINISetting fileIni "Bend" "maxDirection") as integer)catch()
try(bend_limit.triState = (getINISetting fileIni "Bend" "Limit") as integer)catch()
try(bend_upperL_min.value = (getINISetting fileIni "Bend" "upperLimitMin") as
```

```

        integer)catch()
try(bend_upperL_max.value = (getINISetting fileIni "Bend" "upperLimitMax") as
integer)catch()
try(bend_lowerL_min.value = (getINISetting fileIni "Bend" "lowerLimitMin") as
integer)catch()
try(bend_lowerL_max.value = (getINISetting fileIni "Bend" "lowerLimitMax") as
integer)catch()

try(skew_amo_min.value = (getINISetting fileIni "Skew" "minAngle") as integer)catch()
try(skew_amo_max.value = (getINISetting fileIni "Skew" "maxAngle") as integer)catch()
try(skew_dirac_min.value = (getINISetting fileIni "Skew" "minDirection") as integer)catch()
try(skew_dirac_max.value = (getINISetting fileIni "Skew" "maxDirection") as integer)catch()
try(skew_limit.triState = (getINISetting fileIni "Skew" "Limit") as integer)catch()
try(skew_upperL_min.value = (getINISetting fileIni "Skew" "upperLimitMin") as
integer)catch()
try(skew_upperL_max.value = (getINISetting fileIni "Skew" "upperLimitMax") as
integer)catch()
try(skew_lowerL_min.value = (getINISetting fileIni "Skew" "lowerLimitMin") as
integer)catch()
try(skew_lowerL_max.value = (getINISetting fileIni "Skew" "lowerLimitMax") as
integer)catch()

try(taper_amo_min.value = (getINISetting fileIni "Taper" "minAngle") as integer)catch()
try(taper_amo_max.value = (getINISetting fileIni "Taper" "maxAngle") as integer)catch()
try(taper_curve_min.value = (getINISetting fileIni "Taper" "minDirection") as integer)catch()
try(taper_curve_max.value = (getINISetting fileIni "Taper" "maxDirection") as
integer)catch()
try(taper_limit.triState = (getINISetting fileIni "Taper" "Limit") as integer)catch()
try(taper_upperL_min.value = (getINISetting fileIni "Taper" "upperLimitMin") as
integer)catch()
try(taper_upperL_max.value = (getINISetting fileIni "Taper" "upperLimitMax") as
integer)catch()
try(taper_lowerL_min.value = (getINISetting fileIni "Taper" "lowerLimitMin") as
integer)catch()
try(taper_lowerL_max.value = (getINISetting fileIni "Taper" "lowerLimitMax") as
integer)catch()
)
)

rollout abouts "Info" width:400 height:300
(
label info_label01 "ConGen a generative macroScript"
label info_label02 "by Saro Dionyan"
label info_label03 "BOM 2005"
)
conGenFloater = newRolloutFloater "ConGen" 360 700
addRollout abouts conGenFloater
addRollout zDimensions conGenFloater
addRollout ConGen_pos conGenFloater
addRollout ConGen_mod conGenFloater
)--end macroscript

```

ÖZGEÇMİŞ

Doğum tarihi 26.02.1977

Doğum yeri İstanbul

Lise 1988-1996 İstanbul Lisesi

Lisans 1996-2002 Mimar Sinan Üniversitesi Mimarlık Fak.
Mimarlık Bölümü

Yüksek Lisans 2002- Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü
Bilgisayar Ortamında Tasarım. Anabilim Dalı,
Bilgisayar Ortamında Mimarlık Programı

