

**YILDIZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

**DİJİTAL FORM TÜRETİCİ (FROEBEL FORM TÜRETİCİ)
İLE BİR KONUT YERLEŞKESİNİN TASARIM SÜRECİ**

Mimar Gökhan KESKİN

**FBE Mimarlık Anabilim Dalı Bilgisayar Ortamında Mimarlık Programında
Hazırlanan**

YÜKSEK LİSANS TEZİ

Tez Danışmanı: Doç. Dr. Birgül ÇOLAKOĞLU (YTÜ)

İSTANBUL, 2008

İÇİNDEKİLER

	Sayfa
KISALTIMA LİSTESİ	IV
SİMGE LİSTESİ	V
ŞEKİL LİSTESİ	VI
ÖNSÖZ.....	IX
ÖZET	X
ABSTRACT	XI
1 GİRİŞ.....	1
1.1 Amaç.....	3
1.2 Kapsam	3
1.3 Yöntem	3
2 KURAL TABANLI TASARIM.....	4
2.1 Algoritma ve Tasarım.....	4
2.1.1 Algoritma.....	4
2.1.2 Script Dilleri	5
2.1.3 Algoritmik Tasarım	6
2.1.4 Algoritmik Mimarlık	8
2.2 Kural Tabanlı Tasarım Yöntemi Biçim Gramerleri.....	10
2.2.1 Karmaşık Gramerleri	13
2.2.1 Renk Gramerleri	14
2.2.1 Froebel Blokları	14
2.3 Biçim Gramerleri Uygulamaları.....	17
2.3.1 İki Boyutlu Biçim Gramerleri Uygulamaları.....	19
2.3.1.1 El ile Yapılan İki Boyutlu Biçim Gramerleri Uygulamaları	19
2.3.1.2 Dijital İki Boyutlu Biçim Gramerleri Uygulamaları	21
2.3.2.2 Üç Boyutlu Biçim Gramerleri Uygulamaları	22
2.3.2.1 El ile Yapılan 3 Boyutlu Biçim Gramerleri Uygulamaları.....	22
2.3.2.2 Dijital 3 Boyutlu Türetici Gramer Uygulamaları	26
3 NESNE TABANLI ALGORİTMİK TÜRETME MODELİ : FROEBEL FORM TÜRETİCİ.....	30
3.1 Nesne Tabanlı Türetme.....	30
3.2 Tanım.....	32

3.3	Froebel Form Türetici.....	33
3.4	Sentez.....	47
4	GRAMER TABANLI KONUT YERLEŞME TASARIMI.....	48
4.1	Tasarım Sürecini Tasarlama	48
4.2	Scripting.....	48
4.3	Kural Setleri ve Kural Tanımlama.....	49
4.4	Tasarımcı Müdahalesi.....	51
4.5	Form Oluşturma.....	52
4.6	FormunYorumlanması.....	57
5	SONUÇLAR.....	82

KAYNAKLAR

İNTERNET KAYNAKLARI

EKLER

ÖZGEÇMİŞ

KISALTIMA LİSTESİ

BDT	Bilgisayar destekli tasarım
3D	3 boyutlu
3B	3 boyutlu
TAKS	Taban alanı katsayısı
KAKS	Kat alanları katsayısı

SİMGE LİSTESİ

A1	Biçim grameri dizisinin başlangıç biçimi
A2	Biçim grameri dizisinin ikinci biçimi
A3	Biçim grameri dizisinin ikinci biçimi
A4	Biçim grameri dizisinin ikinci biçimi
A1X	A1 biçiminin x düzlemindeki boyutu
A1Y	A1 biçiminin y düzlemindeki boyutu
A1Z	A1 biçiminin z düzlemindeki boyutu
A2X	A2 biçiminin x düzlemindeki boyutu
A2Y	A2 biçiminin y düzlemindeki boyutu
A2Z	A2 biçiminin z düzlemindeki boyutu
A3X	A3 biçiminin x düzlemindeki boyutu
A3Y	A3 biçiminin y düzlemindeki boyutu
A3Z	A3 biçiminin z düzlemindeki boyutu
A4X	A4 biçiminin x düzlemindeki boyutu
A4Y	A4 biçiminin y düzlemindeki boyutu
A4Z	A4 biçiminin z düzlemindeki boyutu
xn1	A2 biçiminin A1 biçiminin x düzleminde ötelenme miktarı
yn1	A2 biçiminin A1 biçiminin y düzleminde ötelenme miktarı
zn1	A2 biçiminin A1 biçiminin z düzleminde ötelenme miktarı

ŞEKİL LİSTESİ

Şekil 2.1	Akış diyagramı	5
Şekil 2.2	Algoritmik tasarım	7
Şekil 2.3	Algoritmik tasarım örneği script arayüzü	7
Şekil 2.4	Algoritmik tasarım (Michael Hansmeyer).....	8
Şekil 2.5	Parametrik mimarlık (Michael Hansmeyer)	9
Şekil 2.6	Biçimler	11
Şekil 2.7	Biçimler arası ilişkiler	11
Şekil 2.8	A1 ile A2 arasındaki ilişki-1	12
Şekil 2.9	A1 ile A2 arasındaki ilişki-2.....	12
Şekil 2.10	A1 ile A2 arasındaki ilişki-3.....	12
Şekil 2.11	A1 ile A2 arasındaki ilişki-4.....	13
Şekil 2.12	Konut (Konur Gönenç).....	13
Şekil 2.13	Müze (Jin Ho Park).....	14
Şekil 2.14	Froebel Blokları	15
Şekil 2.15	Şelale evi (Frank Lloyd Wright)	15
Şekil 2.16	Bloklarla script kullanarak form türetme	16
Şekil 2.17	Buz ışını analizleri	19
Şekil 2.18	Parametrik biçim gramerleri	20
Şekil 2.19	İki boyutlu biçim gramer uygulama örnekleri (Sevkan Harman, Y.T.Ü)	21
Şekil 2.20	İki boyutlu biçim gramer uygulamaları ve üretilmiş örnekler [7] (Y.T.Ü)	22
Şekil 2.21	MIT Terry Knight atölyesi [5] ‘Tarih Müzesi’ Kurallar ve türetmeler (Randy Brown, MIT)	22
Şekil 2.22	MIT Terry Knight atölyesi [5] ‘Tarih Müzesi’ Maketi (Randy Brown, MIT)	23
Şekil 2.23	MIT Terry Knight atölyesi ‘Elementary School Complex’ Tasarım süreci (Michael Brown).....	23
Şekil 2.24	MIT Terry Knight atölyesi ‘Elementary School Complex’ Maket ve vaziyet(Michael Brown, MIT)	24
Şekil 2.25	Gramer tabanlı konut tasarımı(Togay Özkaraduman,YTÜ)	25
Şekil 2.26	Yeni türetilen Malagueira Evleri (Duarte, 2001).....	26
Şekil 2.27	Tasarlanan program ile üretilen formlar(Belinda Torus,YTÜ)	27
Şekil 2.28	3D Architecture Form Synthesizer (Yufei, MIT)	27
Şekil 2.29	QShaper (Tuğrul Yazar,Birgül Çolakoğlu YTÜ)	28
Şekil 2.30	Script ile türetmeler	28
Şekil 2.31	Script ile türetmeler	29
Şekil 2.32	Yorumlama eskizleri	29
Şekil 2.33	Kız Öğrenci Yurdu (Serkan Palabıyık, YTÜ)	29
Şekil 3.1	Froebel form türetici arayüzü	33
Şekil 3.2	Froebel form türetici ile tasarlama.....	35
Şekil 3.3	A1 ile A2 arasındaki ilişki	36
Şekil 3.4	Tanımlanan blok ilişkileri	37
Şekil 3.5	Blok boyutları	37
Şekil 3.6	A1 ile A2 arasındaki ilişki	38
Şekil 3.7	A2 ile A1 arasındaki ilişki	38
Şekil 3.8	Döngü sayısı 5 ile türetilme	38
Şekil 3.9	Üretilen form aşama 1	39
Şekil 3.10	Froebel form türeticide 1. dönüşüm noktasını kullanma	39
Şekil 3.11	Froebel form türeticide 1. dönüşüm noktası blok boyutları	40

Şekil 3.12	Froebel form türeticide 1. dönüşüm noktası blok ilişkileri	40
Şekil 3.13	Üretilen form aşama 2	41
Şekil 3.14	Froebel form türeticide 2. dönüşüm noktasını kullanma	41
Şekil 3.15	Froebel form türeticide 2. dönüşüm noktası blok boyutları	42
Şekil 3.16	Froebel form türeticide 2. dönüşüm noktası blok ilişkileri	42
Şekil 3.17	Üretilen form aşama 3	43
Şekil 3.18	Üretilen form aşama 3	43
Şekil 3.19	5 dönüşüm noktası kullanılarak üretilen form	44
Şekil 3.20	5 dönüşüm noktası kullanılarak üretilen form	45
Şekil 3.21	Form türetme denemesi-1,2	45
Şekil 3.22	Form türetme denemesi-3,4	46
Şekil 3.23	Form türetme denemesi-5,6	46
Şekil 3.24	Form türetme denemesi-7,8	46
Şekil 4.1	Konut yerleşmesinin tasarlanacağı arsa	48
Şekil 4.2	Seçilen bloklar	49
Şekil 4.3	Bloklar arası ilişki seçenekleri	50
Şekil 4.4	1 dönüşüm noktası kullanılarak türetilen seçenekler ve seçilen form	51
Şekil 4.5	2 dönüşüm noktası kullanılarak türetilen form-1	52
Şekil 4.6	2 dönüşüm noktası kullanılarak türetilen form-2	53
Şekil 4.7	2 dönüşüm noktası kullanılarak türetilen form-3	53
Şekil 4.8	2 dönüşüm noktası kullanılarak türetilen form-4	54
Şekil 4.9	2 dönüşüm noktası kullanılarak türetilen form-5	54
Şekil 4.10	2 dönüşüm noktası kullanılarak türetilen form-6	55
Şekil 4.11	2 dönüşüm noktası kullanılarak türetilen form-7	55
Şekil 4.12	2 dönüşüm noktası kullanılarak türetilen form-8	56
Şekil 4.13	2 dönüşüm noktası kullanılarak türetilen form-9	56
Şekil 4.14	2 dönüşüm noktası kullanılarak türetilen form-10	57
Şekil 4.15	Seçenek 10'a ait görüşler ve perspektif	58
Şekil 4.16	Seçenek 10'a ait perspektif	59
Şekil 4.17	Forma müdahaleler (Seçenek 10)	60
Şekil 4.18	Üst görünüm (Seçenek 10)	60
Şekil 4.19	Zemin kat planı (Seçenek 10)	61
Şekil 4.20	Üst kat planı (Seçenek 10)	61
Şekil 4.21	Süreç perspektifleri (Seçenek 10)	62
Şekil 4.22	Konut perspektifleri (Seçenek 10)	63
Şekil 4.23	Görünümleri karşılaştırılması (Seçenek 10)	64
Şekil 4.24	Seçenek 1'e ait görüşler ve perspektif	65
Şekil 4.25	Seçenek 1'e ait perspektif	66
Şekil 4.26	Forma müdahaleler (Seçenek 1)	67
Şekil 4.27	Üst görünüm (Seçenek 1)	67
Şekil 4.28	Planlar (Seçenek 1)	68
Şekil 4.29	Süreç perspektifleri (Seçenek 1)	69
Şekil 4.30	Konut perspektifleri (Seçenek 1)	70
Şekil 4.31	Görünümleri karşılaştırılması (Seçenek 1)	71
Şekil 4.32	Seçenek 5'e ait görüşler ve perspektif	72
Şekil 4.33	Seçenek 5'e ait perspektif	73
Şekil 4.34	Forma müdahaleler (Seçenek 5)	74
Şekil 4.35	Üst görünüm (Seçenek 5)	74
Şekil 4.36	Zemin kat planı (Seçenek 5)	75

Şekil 4.37 Zemin kat planı (Seçenek 5)	75
Şekil 4.38 Süreç perspektifleri (Seçenek 5).....	76
Şekil 4.39 Konut perspektifleri (Seçenek 5).....	77
Şekil 4.40 Görünüşleri karşılaştırılması (Seçenek 5)	78
Şekil 4.41 Konut yerleşkesi perspektif	79
Şekil 4.42 Konut yerleşkesi görünüşleri	80
Şekil 4.43 Konut yerleşkesi perspektif	81

ÖNSÖZ

Yıldız Teknik Üniversitesi Mimarlık Bölümü Bilgisayar Ortamında Mimarlık Yüksek Lisans programında hazırladığım tez, bundan sonraki mimarlık hayatımda kullanabileceğim bir tasarım yöntemi geliştirmemi sağlamıştır. Bana ve bu tezin oluşumuna katkısı bulunan Bilgisayar Ortamında Mimarlık Yüksek Lisans Programında katıldığım tüm derslerin değerli öğretim görevlilerinin her birine, bu süreç içinde destek ve hoşgörülerini esirgemeyen eşime, aileme ve işyeri arkadaşlarıma teşekkür ederim.

Tez çalışmasının oluşumunda en büyük katkıya sahip olan sayın Doç.Dr. Birgül Çolakoğlu'na, bana gösterdiği hoşgörü, verdiği güven ve destekten ötürü özellikle teşekkür ederim.

İstanbul, Ağustos 2008

Gökhan KESKİN

ÖZET

Bilgisayarın mimarlar tarafından kullanımının artması ile birlikte, mimarların bilgisayarları tasarım sürecinde de kullanma ihtiyaçları çeşitli yazılımlarla birlikte mümkün olmaya başlamıştır. Mimarlar ya da tasarımcılar, bilgisayarın hesaplamalı gücünden yararlanarak, el yordamı ile yapılması mümkün olmayan ya da oldukça zor olan yöntemler geliştirebilmektedirler. Mimarların yazılımları geliştirmeleri zor olsa da, yazılımların sundukları yazılımlara müdahale etme ve istediği arayüzü geliştirmeyi sağlayan script dillerini anlama ve uygulamaları daha kolay olmaktadır. Bu yöntem diğer programlama dillerine oranla oldukça anlaşılırdır. Script mantığını öğrenip çözümlenebilen tasarımcılar, problemlerini çözecek algoritmaları geliştirmekte zorlanmayacaklardır. Tabii ki burada algoritmik düşünce ve algoritmik tasarım kavramları önem kazanmaktadır. Tasarımcı problemin çözümünde izleyeceği yolu kendisi belirleyip algoritmasının tasarlayacaktır. Bu da tasarım sürecinin tasarlanması anlamına gelmektedir. Benim yaptığım araştırma ile Froebel blokları ile biçim gramerleri kurallarını kullanarak form üretmeyi amaçlayan bir algoritma geliştirilmiştir. Froebel Form Üretici adını verdiğimiz bu arayüz ile her türlü form arayışı yapılabilmektedir. Çalışmada bir konut yerleşmesinin tasarım süreci ele alınmıştır.

Araştırmanın ikinci bölümünde kural tabanlı tasarım yöntemlerinden biçim gramerleri geniş bir şekilde ele alınmaktadır. Araştırma kapsamında geliştirilen eklenti program olan Froebel Form Üretici üçüncü bölümde ele alınıp kullanımı anlatılmakta ve elde edilen bazı örnek formlar sunulmaktadır. Dördüncü bölümde ise Froebel form üretici kullanılarak konut yerleşmesinde uygulanacak konut formlarının elde edilme süreçleri anlatılmaktadır.

Anahtar Kelimeler: Bilgisayar Destekli Tasarım, Algoritmik Mimarlık, Kural Tabanlı Tasarım, Biçim Gramerleri, Froebel Blokları, Genetik Algoritmalar, Evrimsel Mimarlık

ABSTRACT

Nowadays, the need of using softwares during design process by architects and designers become to be possible with the developementes in computer technologies. Architects and desginers can improve some new methods with using the computational capacity of computers which are impossible or difficult to do maually. It can be very hard to create a software for designers, but with the script languages of the softwares which makes possible to interfere the software and to create the special interfaces for users, it becomes to be more easier to understand and practise. The code of a script is easier than to understand a software code. It won't be hard to design the algorithms to solve the problems for architects who can learn the scripting rules. Of course, the notions of algorithmic thinking and algorithmic design becomes important. Architects will assign the solution of the problem and design the algorithm. And this means the design of design process. With my research, an algorithm is being designed to generate forms which uses the froebel blocks with shape grammar rules. With the interface that is called Froebel Form Generator, we can make form findings for everything. In my work, the design process of a housing estate is handled.

The second part of the research is about the rule based design techniques. In this part, the main subject shape grammars are explained. In the third part, Froebel form generator, which is designed during the research, is explained and some examples generated with it are shown. And in the fourth part, the stages of to create the houses of the housing estate with froebel form generator is defined.

Keywords:

Computer Aided Design, Algorithmic Architecture, Rule Based Design, Shape Grammars, Froebel Blocks, Genetic Algorithms, Evolutional Architecture

1. GİRİŞ

Bu çalışma ile bilgisayar teknolojileri ile paralel bir şekilde gelişim gösteren bilgisayar destekli tasarım tekniklerinden kural tabanlı tasarım yöntemleri üzerinde durulacaktır.

Bilgisayarın mimarlık alanındaki kullanımı BDT programları ile birlikte yaygınlaşırken, bilgisayar daha çok bir mimari sunum tekniği aracı olarak gelişmeye başladı. Yazılımların yetersiz olmaları, henüz tasarımcının hayal ettiklerini gerçekleştirecek donanım ve alt yapıya sahip olmaması, BDT programlarını karşı bir önyargının oluşmasına neden oldu. Zaman içinde yazılımların gelişmesi, ara yüzleri ve sunum teknikleri üst düzey programların gelişmesi ile birlikte bilgisayara karşı olan bu önyargı kayboldu ve bilgisayar mimarlık disiplininin vazgeçilmez bir ögesi oldu. Bilgisayar artık tasarımı kolaylaştıran, hızlandıran ve hatta tasarımı etkileyen bir tasarım aracı, tekniği haline geldi.

Bilgisayar bir tasarım aracı olarak mimarlık içindeki yerini alırken, bir yandan da biçim gramerleri, genetik algoritmalar gibi kural tabanlı tasarım yöntemleri üzerine yapılan araştırmalar bilgisayar ortamında çalışılmaya başlandı. Bu çalışmalar beraberinde yeni tartışmaları ve yeni araştırma alanlarını getirdi. Tasarımcının bilgisi, yeteneği, hisleri, var oluşsal özellikleri ve fiziksel çevrenin etkenleri ile gelişen geleneksel tasarım yöntemlerinde tasarımcı, önsezileri ile tasarım ve ürün üzerinde mutlak kontrole sahiptir. Bilgisayar destekli tasarımın ise henüz yeni kullanılmaya başlanması ve bilgisayar destekli tasarım araçlarını kullanmadaki zorluk ve karmaşıklık, bilgisayar destekli tasarımda tasarımın kontrolünün bilgisayara bırakıldığı gibi bir his uyandırabilmektedir. Yaratıcılığın insan gelişimi ile oluştuğu ve her tasarımcının kendine özgü sanatsal stilinin olduğu düşüncesine karşın, bilgisayar destekli tasarımda kontrolün bilgisayara bırakıldığı düşüncesi, estetik uyumu, ahengi, ritmi, dengeyi ve işlevselliği yakalama konusunda bazı hayal kırıklıklarının oluşabileceği gibi endişelere neden olabilmektedir. Bilgisayar destekli tasarım araçlarını ve yazılımları kullanan tasarımcılar, gerek yazılımların yetersiz olması, gerekse bilgilerinin eksik olması nedeniyle bu yöntemler ile yeni formlar yaratmanın bir rutine dönüşeceği izlemine de kapılabilmektedirler. Oysaki tasarım uzayını zengin tutmak ve form çeşitliliğini sağlamak kullanıcının kendi kontrolüyle gerçekleşebilmektedir.

Bilgisayar destekli tasarımda da geleneksel tasarım yöntemlerindeki tasarımı etkileyen faktörler aynı şekilde tasarıma etki eder ve bu süreçte bilgisayar destekli tasarım araçları ile

tasarımcının iletişimi önem kazanır. Tasarımcının araya girip müdahale şansı bilgisayar destekli tasarım yönteminin de kontrolünü sağlar. Tasarımcının hangi kuralın nerde, ne zaman, nasıl uygulayacağını, işleyişin ne zaman başlayacağı, ne zaman biteceği gibi seçenekleri kontrol şansı bulunmaktadır. Bu sayede kararlar tasarımcı tarafından verilmekte tasarımın ve yöntemin kontrolü ve akışı yönlendirilmektedir. Bu aşamaların kontrollerinde tasarımcının öngörülerini ve sezileri önem kazanmaktadır. Eğer tasarımcı hangi formların üreyebileceğini öngörebiliyor ve yazılımı bu doğrultuda kontrol edebiliyorsa, çalışmanın niteliği ve zenginliği de o ölçüde artabilecektir.

Bilgisayar destekli tasarım ile önceden kestirilmesi imkânsız olan bütün potansiyel formlar ve biçimler görülebilmekte, oluşabilecek sürprizlerle tasarıma farklı boyutlar kazandırılabilir. Tasarım uzayını zengin tutmak için yazılımı etkin kullanarak, formlarda değişimler, dönüşümler oluşturularak, kendiliğinden gelişimi sağlayacak müdahaleleri belirli süreç ve noktalarda uygulamak gerekmektedir. Yazılımın bu dönüşüm ve değişim noktalarını doğru zaman ve süreçlerde alması gerekmektedir. Bu kararlar ve öngörüler ise daha çok tasarım kararları ve daha çok yaratıcılık gerektirmektedir. Bilgisayar destekli tasarım yazılımlarının gelişmesi, kullanımının ve kontrolünün de kolaylaşmaya başlaması ile yaygınlaşmaya başlayan biçim gramerleri ve genetik algoritmalar gibi ana başlıklar altında ele alınan üretken tasarım metotları, tasarım uzayını zenginleştirmekte ve sadece sonuç ürünü zenginleştirmekle kalmayıp metotları da etkilemektedir.

Evrimsel yöntemlerle geliştirilen genetik algoritmalar ile ise sanal evrim içerisindeki mimarlığın yeri araştırılıyor. Bu çalışmaların mimari tasarımdaki yeri ve kullanılabilirliği yeni bir alan oluşturmaktadır. Evrim teorisindeki evrimsel süreçler düşünüldüğünde farklı oran ve süreçlerde oluşan mutasyonlarla daha geniş üretken bir topluluk oluşmakta, bu da sonuç ürünün çeşitliliğini artırmaktadır. Bu yöntemler mimari tasarım sürecinde uygulandığında tasarımcının, değişim, farklılaşma sürecinde formun ne yönde mutasyonlara uğrayacağını, hangi genlerin sonraki genlere iletilip ileilmeyeceğini ve değişip değişmeyeceğini saptaması, formu önsezileri ve öngörülerini ile kontrol ederek türetmesi gerekmektedir. Sanal evrim içerisinde yerini bulan evrimsel mimarlık, tasarımcıyı heyecanlandıran, tasarımcının üretkenliğini artıran, tasarımcıyı zorlayan yönleriyle, zengin tasarım uzayı ve sürprizlerle dolu sonuçlarıyla mimarlık alanının en geniş araştırma alanlarından biri haline gelmektedir.

1.1. Amaç

BDT programlarının ofislerde ve öğrenciler arasındaki kullanımının son derece yaygın olmasına karşın, üretken tasarım metotlarını oluşturan unsurlardan biçim gramerleri ve genetik algoritmalar gibi tasarım yöntemleri, gerek üniversite öğrencilerince, gerekse ofislerde yeterince kullanılmamaktadır. Bunun nedeni ise, bu tasarım tekniklerinin bilişim teknolojilerine paralel gelişmesi ve bilgisayar mühendisliği ile iç içe olmasıdır. Öğrenci ve ofislerdeki kullanımındaki güçlükler düşünüldüğünde; üretken tasarım tekniklerinin öneminin ve getirilerinin çok iyi anlatılması gerekir ve üniversite eğitim sürecinde ilgili derslere daha çok yer verilebilir. Bilişim teknolojilerindeki gelişmeler ile etkileşim içinde olan mimar, üretken tasarım teknikleri ile ilgili altyapı ve bilgiye sahip olduğunda kendini geliştirmesi ve gelişmeleri takip etmesi daha kolay olacaktır. Aksi takdirde üretken tasarım teknikleri tasarım sürecinde ihmal edilmeye devam edilecektir. Bu bağlamda, araştırma ile bir konut yerleşmesinin tasarım süreci, maxscript kullanılarak tasarlanan eklenti program ile geliştirilecektir. Süreç ve neticesinde alınan sonuçlar ile üretken tasarım tekniklerinin mimari tasarım sürecinde kullanılması değerlendirilecektir.

1.2. Kapsam

Araştırma bilgisayar destekli tasarımın kural tabanlı tasarım teknikleri üzerinedir. Kural tabanlı tasarım tekniklerinden biçim gramerleri ele alınmıştır. Yapılan deneysel çalışmada Froebel blokları ile form türetilirken biçim gramerleri kuralları kullanılmıştır. Form üretmek için tasarlanan algoritmaların kontrolünü ve kullanımını kolaylaştırmak için maxscript ile bir arayüz oluşturulmuştur. Biçim gramer teknikleri ve bu konu ile ilgili uygulama örneklerine yer verdikten sonra ise bir konut tasarımının biçim gramerleri ile tasarım süreci incelenmiştir. Bu tasarım sürecinde froebel blokları, scripting, algoritmik tasarım kavramları ele alınmıştır.

1.3. Yöntem

Araştırmaya kural tabanlı tasarım yöntemlerinin temelini oluşturan algoritma kavramının tanımı ve tasarımda algoritma mantığının anlatılması ile bir giriş verildikten sonra araştırmanın ana konusu olan biçim gramerleri detaylı bir şekilde ele alınmaktadır. Daha sonra ise; biçim gramer kuralları ve froebel blokları kullanılarak, mimari tasarım sürecinde değerlendirilmek üzere, 3 boyutlu form eskizleri elde etmeyi hedefleyerek maxscript ile geliştirilen froebel form üretici arayüzü anlatılmaktadır. Araştırma, froebel form üretici

kullanılarak bir konut projesinin tasarım sürecinin irdelenmesi ve sonuçlarının değerlendirilmesi ile sona etmektedir.

2. KURAL TABANLI TASARIM

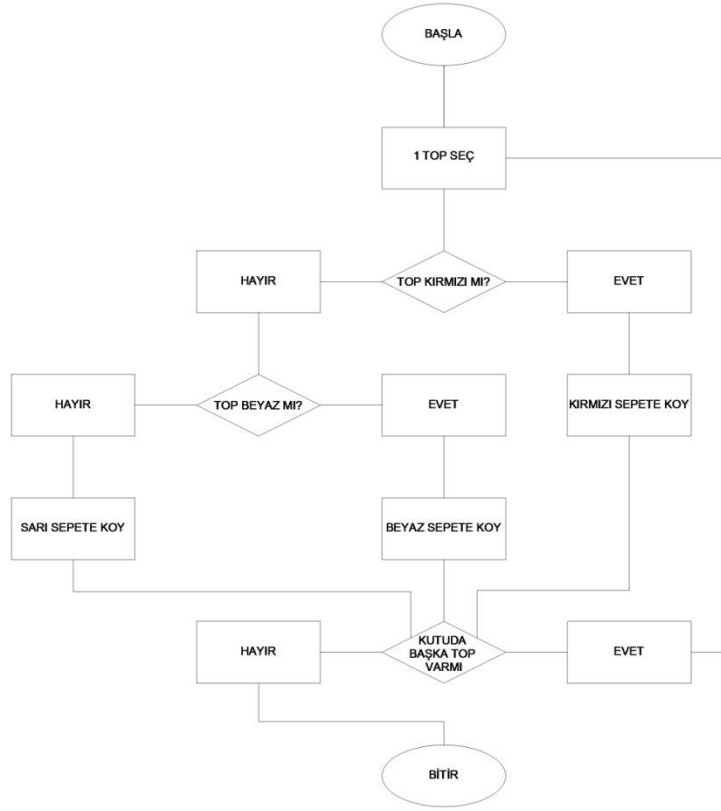
2.1. Algoritma ve Tasarım

2.1.1. Algoritma

Algoritma, matematikte ve bilgisayar biliminde kullanılan, bir işi yapmak için bir başlangıç durumundan başladığında, açıkça belirlenmiş bir son durumda sonlanan, sonlu işlemler (adımlar) kümesidir. Algoritmalar bilgisayarlar tarafından işletilebilirler. Algoritma kelimesi, Özbekistan'ın Horasan kentinde doğmuş Ebu Abdullah Muhammed bin Musa el Harezmi isimli Fars matematikçinin adından gelir. Batılılar, *el Harezmi* (Al-Khwārizmī) sözcüğünü telaffuz edemedikleri için terim bu şekilde kalmıştır [1].

Algoritma, bilgisayar bilimlerinde, herhangi bir sorunun çözümü için izlenecek yol anlamına gelmektedir. Algoritmalarda çözüm için yapılması gereken işlemler alternatif yoruma izin vermeksizin net bir şekilde ifade edilir. Algoritma ile bilgisayara hangi çevre biriminden girileceği, problemin nasıl çözüleceği, hangi basamaklardan geçirilerek sonuç alınacağı, sonucun nasıl ve nereye yazılacağı sözel olarak ifade edilmesi olarak da tanımlanabilir.

Algoritmaların simge ya da sembollerle ifade edilmiş şekline "akış diyagramları" adı verilir. Akış diyagramlarının algoritmadan farkı, adımların simgeler şeklinde kutular içine yazılmış olması ve adımlar arasındaki ilişkilerin ve yönünün oklar ile gösterilmesidir. Aşağıdaki Şekil 2.1'de bir kutu içinde bulunan kırmızı beyaz ve sarı renkteki topların ayrıştırılarak kırmızı beyaz ve sarı renkteki üç sepete bölünmesini çözümlen bir algoritmanın akış diyagramı yer almaktadır.



Şekil 2.1 Akış diyagramı

2.1.2. Script Dilleri

Script dilleri; C++, Pascal, vb. gibi sistem programlama dillerinden farklı olarak özel görevler için geliştirilmiş yapılardır. Sistem programlama dilleri ile yeni sistemler üretilir. Script dilleri ise, mevcut bir dilin alt bütünlerini kullanarak bu bütünleri bağlar. Script dilleri karmaşık algoritma ve veri yapıları için kullanılmaz. Sistem programlama dillerindeki sözdizim kuralları oldukça önemli ve zordur. Script dillerinde ise uygulamayı hızlandırmak ve kolaylaştırmak için sözdizim kuralları sadeleştirilmiştir. Script dilinde elemanların nasıl kullanılacağını ve bir araya geleceğini gösteren tanımlanmış kalıplar yoktur [2]. Sistem programlama dillerindeki zorluk düşünüldüğünde, script dillerinin getirdiği kolaylıklar ile mimarlar ve tasarımcılar, problemlerinin çözümleri için yeni yöntemler geliştirme imkânı bulabilmektedirler.

Bilgisayar destekli tasarım yazılımları da yeni çıkan sürümlerinde script dillerini yazılımlarına eklemekte ya da var olan dilleri geliştirmektedirler. AutoLisp, Maya Encoding Language MEL, MaxScript, RhinoScript vb. gibi script dilleri ile tasarımcılar kendi tasarım

yöntemlerini geliştirme şansını bulmaktadırlar. BDT yazılımlarının, mimari sunum tekniği olarak kullanılmasının yanı sıra tasarım aracı olarak da kullanılmaları script dilleri ile yaygınlaşmaya başlamaktadır.

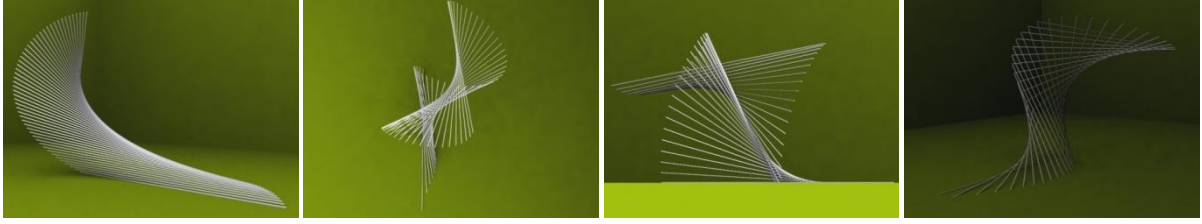
2.1.3. Algoritmik Tasarım

Geleneksel tasarım yöntemlerinde bilgisayarın etkin bir biçimde kullanılması tasarımcının yazılımlar ile olan iletişimine bağlıdır. Bu durumda bilgisayar ve yazılımlar kullanıcının iletişimine göre tasarıma yardımcı olur, görselleştirme ve sunuş zenginlikleri sağlayarak sonuç ürünün temsilini gerçekleştirirler. Bilgisayar ile doğrudan etkileşim içinde olan tasarımcı, süreci doğrudan kontrol eder ve çıkacak sonuç ürüne de hâkimdir.

Algoritmik tasarım ise kullanıcıya yeni, karmaşık kavramlar ve sorunlar getirmektedir. Algoritmayı belli bir problemin çözümü için izlenecek yolun açık tasarlanması olarak özetlemiştik. Algoritmik tasarım ise bu durumda tasarımın da tasarlanması gibi bir anlam taşımaktadır. Tasarım yönteminin ve sürecinin de tasarlanması anlamına da gelmektedir. Mimari tasarım sürecinde bir yapının tasarımı için tek tip bir algoritma veya program yoktur. Tasarlanacak algoritma veya programlar, probleme, tasarımcıya ve diğer etkenlere göre değişecektir. Algoritmik tasarımda süregelen bir değişim vardır.

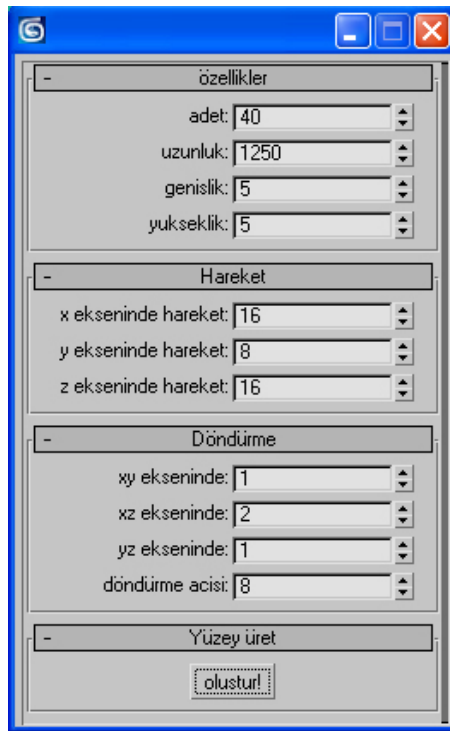
İlk bakışta karmaşık ve zor gibi gözükse de bu yöntem, bilgisayar destekli tasarım araçlarını ve gerekli yazılımları kullanabilen tasarımcıların üretkenliğini artıracaktır. Sürecin stratejilerini ve yöntemini tasarlayan tasarımcı, beklenmedik, öngörülme-yen sonuçları elde etme, tasarıma farklı boyutlar ve zenginlikler kazandırma şansını bulacaktır. Algoritmalar karmaşık sistemleri çözümlenmek veya tasarlamak, problem veya sorunun çözümünü zenginleştirmek, yaratıcılığı artırmak ve yeni yöntemler geliştirmek için kullanılmaktadır.

Son dönemlerde bilgisayar destekli tasarım yöntemlerini etkin bir biçimde kullanmak ve gerekli yazılımları tasarlamak için bilgisayar mühendisi olmaya veya bilgisayar mühendisinden yardım almaya gerek kalmamıştır. Bazı yazılımlar kullanıcılara sundukları yeni yöntemler ile tasarımcının geleneksel yöntemlerin dışında farklı tasarım yöntemleri ile tanışmasına olanak sağlamaktadır. Kullanıcılar kendi istediği işlevleri yerine getirecek arayüzleri script dilleri ile tasarlayabilmektedirler. Bu dilleri anlamak için yazılımcı olmak gerekmemektedir. Basitleştirilmiş kodlamalar ile kullanıcı sadece algoritmaları tasarlayıp dilediği tasarım arayüzünü programa ekleyebilmektedir.



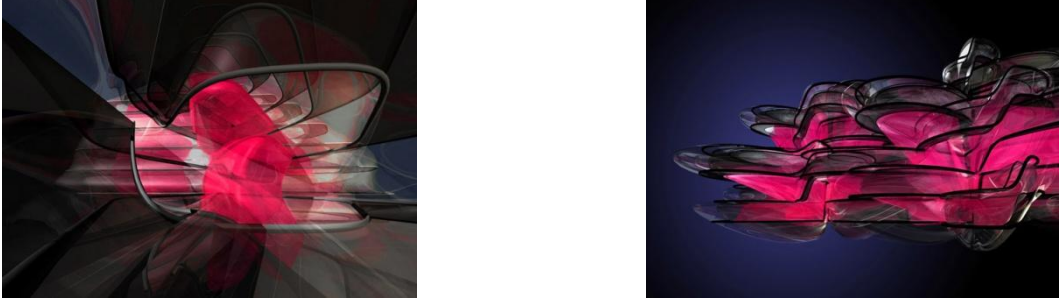
Şekil 2.2 Algoritmik tasarım

Şekil 2.2’de yer alan resimlerde tasarlanan formlar Y.T.U Bilgisayar Ortamında Tasarım Bilim Dalı, Bilgisayar Ortamında Mimarlık Yüksek Lisans Programı kapsamındaki Bilişim Kavramları dersinde yüzey türetmek amacıyla 3dsMax programının script dili olan Maxscript ile tarafımdan oluşturulan algoritma ve Şekil 2.3’de görülen arayüz ile türetilmiştir. Algoritma, bir başlangıç formunun ötelenmesi ve döndürülmesi gibi Şekil 2.3’de görülen basit parametrelerin girilmesiyle oluşmaktadır. Arayüz, yukarıdaki türetmelerin dışında, girilen parametrelerin değiştirilmesine bağlı olarak birçok form-yüzey türetmektedir. Görüldüğü üzere yukarıdaki formların el yordamıyla ya da bilgisayar programları ile türetilmesi çok zaman ve uğraş gerektirecekken tasarlanan algoritma ve arayüz ile süreç ve uğraş azaltılıp sonuç ürün seçeneği zenginleşmektedir.



Şekil 2.3 Algoritmik tasarım örneği script arayüzü

Aşağıdaki Şekil 2.4'te Michael Hansmeyer'in Algoritmik mimarlık başlığında yaptığı türetmelerin sonuç ürünleri bulunmaktadır.



Şekil 2.4 Algoritmik tasarım(Michael Hansmeyer [3])

2.1.4. Algoritmik Mimarlık

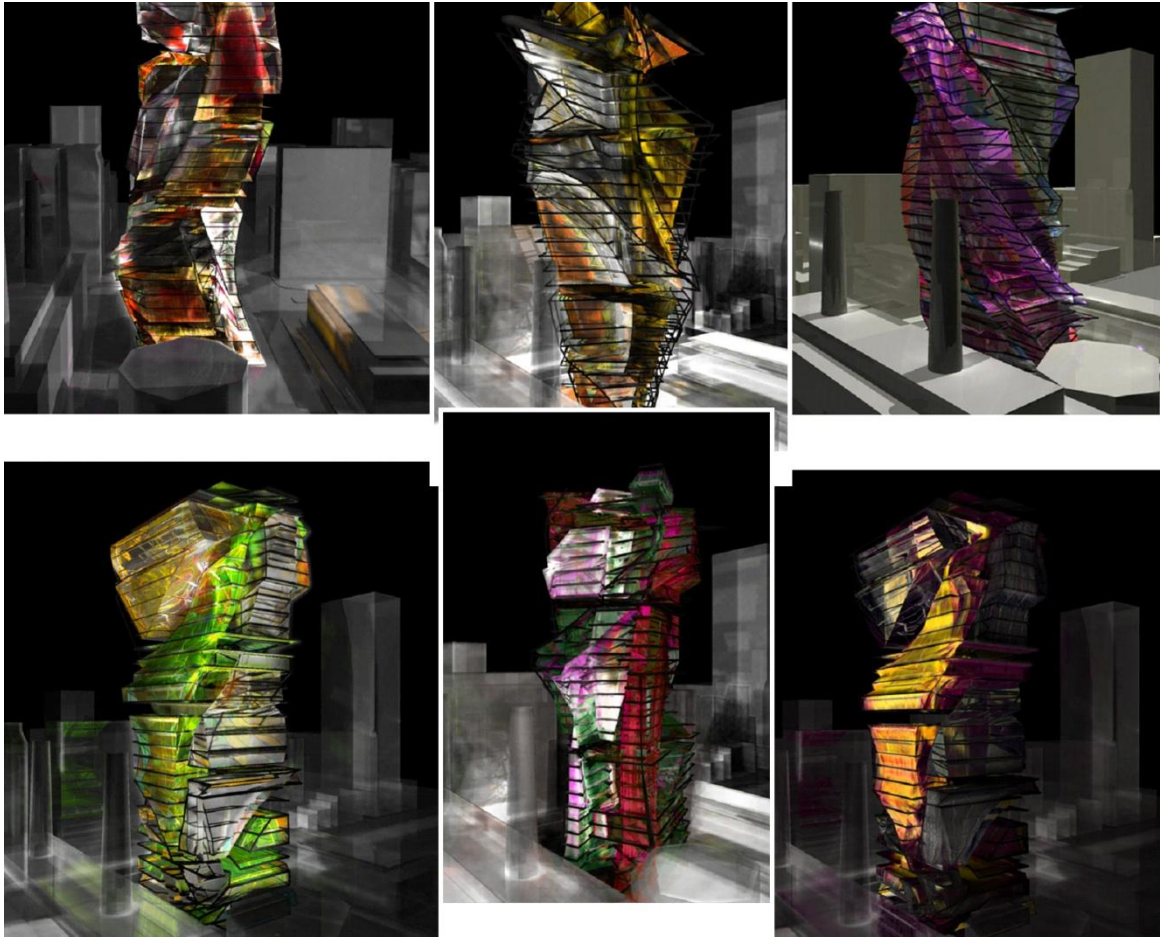
Mimarlar geleneksel kullanımlarda yazılımın imkân verdiği uzay içerisinde tasarım yapabilirler. Yazılımın kurgulanmış mantığına müdahale ederek tasarımcıya yazılımı manipüle etme imkânı sağlayan 3DSMAX, MAYA, CATIA gibi CAD yazılımlarının script dilleri ile mimarların yazılımları tasarlamaları söz konusu olmaktadır. Bu diller sayesinde mimar; bilgisi, zekâsı ve yaratıcı niteliği ile bilgisayarın hesaplamalı yöntemlerdeki üstünlüğünü birleştirerek forma daha çok müdahale etme şansı bulmaktadır.

Yazılımlar tarafından sunulan script dilleri ile mimari programların sunduklarının ötesinde sonuçlara ulaşmayı hedefleyen mimarın, algoritmik düşünce mantığını iyi kavraması gerekmektedir. Ne yapmak istediğini bilmeli, süreci ve yöntemi iyi tasarlamalıdır. Öncelikle kistaslarını; tasarım verilerini, çevresel girdi ve çıktıları, sınırlarını ve diğer parametreleri belirlemeli, daha sonra algoritmanın akış diyagramını tasarlamalıdır. Algoritmalar sadece bilgisayar ve mantıksal işlemler kümesi değildirler. Aynı zamanda felsefesi ve teorik bir altyapısı vardır ve artistik kaygılara dayanan ihtiyaçlar nedeniyle gerek duyulmaktadır. Tasarımcının algoritmik mimari tasarım sürecinde, farklı ve zor olanı yakalama içgüdüsüyle kendisini ve sınırlarını zorlaması, sonuç ürünleri zenginleştirmekte ve mimarın üretkenliğini artırmaktadır.

Algoritmik mimarlık ile bir sorunun sınırlı ya da sınırsız, mümkün olan bütün potansiyel çözüm yolları elde edilebilir. Sürecin ve yazılımın kontrolü mimarın elinde olmasına rağmen mimarın önceden öngöremediği sürpriz ve heyecan verici sonuçlara da varılabilmektedir.

Algoritmanın tasarımı ve parametrelerin niteliğine göre süreç, mimarın tümüyle kontrolünde olduğu gibi rastgelelik içerebilecek parametrelerle gelişigüze de bırakılabilir. Ya da evrimsel yöntemlerle yazılımın kendi içinde bir sistem ile kontrollü ya da kontrolsüz bir biçimde işlemesi sağlanabilir. Kısacası geleneksel yöntemlerde, bir problemin çözümüne pratik yöntemler bulmak için kullanılan algoritmalar, mimari tasarım sürecinde de pratik yöntemler bulmak için kullanılmasının ötesinde insan aklının hayal gücünün ve yaratıcılığının sınırlarını zorlayacak yöntemlerin iskeletini oluşturmaktadır.

Algoritmik mimarlığa bir örnek ise Hansmeyer 'in [3] çeşitli parametrelerle formu manipüle ettiği, Şekil 2.5'te türetmeleri yer alan çalışmadır. Hansmeyer tasarım yöntemi sabit olan ve değişen parametreleriyle olası varyasyonları üretmeyi sağlayan bir algoritma tasarlamıştır. Tasarladığı algoritma formun sonsuz olasılıkta seçeneklerini oluşturabilmektedir. Bu formların tümünün biçimsel kuralları benzerdir.



Şekil 2.5 Parametrik mimarlık (Michael Hansmeyer[3])

2.2. Kural Tabanlı Tasarım Yöntemi Biçim Gramerleri

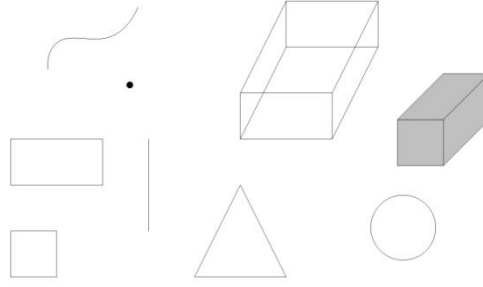
Geleneksel ve bilgisayar destekli tasarım yöntemlerinin bir alanı olan kural tabanlı tasarım yöntemleri, özellikle bilişim alanındaki hızlı gelişmeler ile gittikçe genişleyen, yeni araştırma ve tartışma alanları eklenen bir konu haline gelmiştir. Kural tabanlı tasarım yöntemlerinden en çok kullanılan ise, biçim ve form üretmeye dayalı olan biçim gramerleridir. Teknolojilerin ve dolayısı ile yazılımların gelişmesiyle birlikte, yeni kural tabanlı tasarım yöntemleri üzerinde araştırma ve tartışmalar yapılmaktadır. Çalışmada biçim gramerleri yöntemleri kullanılacağından bu yöntem üzerinde durulacaktır.

Biçim gramerleri biçim kompozisyonları üreten kural tabanlı bir tasarım yöntemidir. İlk olarak Stiny ve Gibs tarafından 70'li yıllarda geliştirilmeye başlanmıştır. Tanımlayıcı ve türetici olan bu metot ile biçimsel temsil dili kullanarak tasarım yapılmaktadır. Biçim kurallarının oluşturulduğu elemanlar nokta, çizgi, yüzey ve masiftir. Biçimler arası ilişkiler ve türetmeler yapılırken kullanılan aritmetik biçim operasyonları, ekleme, çıkarma, kaydırma, öteleme, döndürme gibi mekânsal dönüşümlerdir. İlk dönem biçim grameri çalışmaları analiz ağırlıklıdır. Konu ile ilgili olarak yapılan ilk çalışma Stiny'nin 1977 de yazdığı "Ice-Ray: a note on the generation of Chinese lattice designs" isimli makalesidir. Bu çalışmada Çin buz deseni kompozisyonları beş basit kural içeren gramer ile anlatılmış, gramer kuralları kullanılarak mevcut çin buz desenlerinin tasarım kuralları elde edilmiş ve aynı tarzda sayısız tasarımın yapılabileceği gösterilmiştir. Buna benzer çalışmalar İslam süsleme motiflerinin ve Selçuklu halı desen gramerlerinin çıkarılması şeklinde de uygulanmıştır. Bu çalışmalarda analiz yapılarak tasarım kuralları çıkarılmakta ve bu kurallarla dönüşümler yapılarak yeni tasarım kuralları geliştirilmektedir. Geliştirilen bu yeni tasarım kuralları ile de yeni desenler, formlar türetilmekte, üretkenlik artırılmaktadır (Çolakoğlu 2004).

Biçim gramerlerinin görsel tasarım temsilleri arasında yer alması, tasarımcının formu ve tasarım sürecini kontrol edip kavramasını kolaylaştırmaktadır. Biçim gramerleri algoritmik tasarım yöntemidir. Bilgisayar programları kullanılarak oluşturulan biçim algoritmaları ile sonsuz sayıda biçim kompozisyonları üretilebilmektedir.

Biçim gramerleri bir üretken tasarım metodudur. Türetici gramerlerde sayılar ve karakter dizileriyle oluşan kural setleri yine bir türetici gramer olan biçim gramerlerinde biçimler ve biçimler arası ilişkiler ile tanımlanır. Biçimler nokta çizgi yüzey masif gibi 2 ve 3 boyutlu geometrik elemanlardan oluşur (Çolakoğlu 2004).

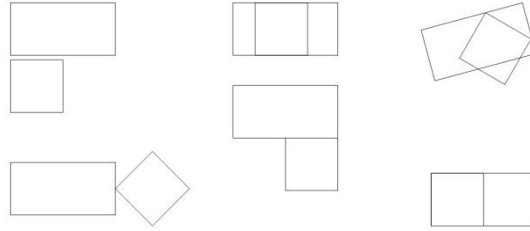
Biçimler



Şekil 2.6 Biçimler

Biçim gramerlerinde bir başlangıç biçimi, ondan türeyen veya diğer biçimlerle oluşan biçim kelimeleri ve biçimlerin arasındaki ilişkileri tanımlayarak oluşturduğumuz kurallar ve kural setleri vardır. Biçimlerin birbirleri ile olan ilişkileri hem ikinci hem de üçüncü boyutta tanımlanabilir.

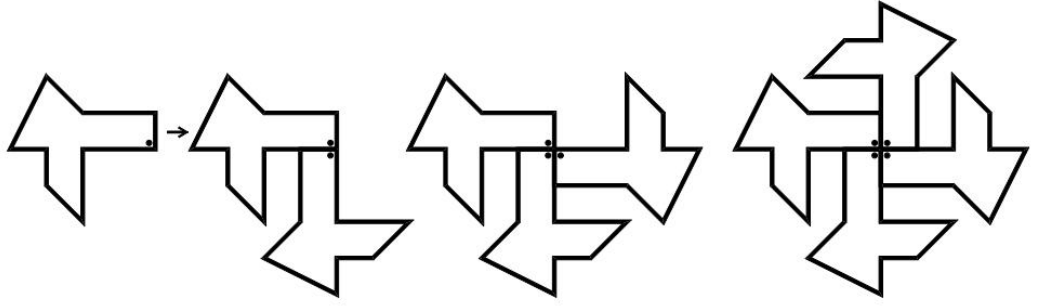
Biçimler Arası İlişkiler



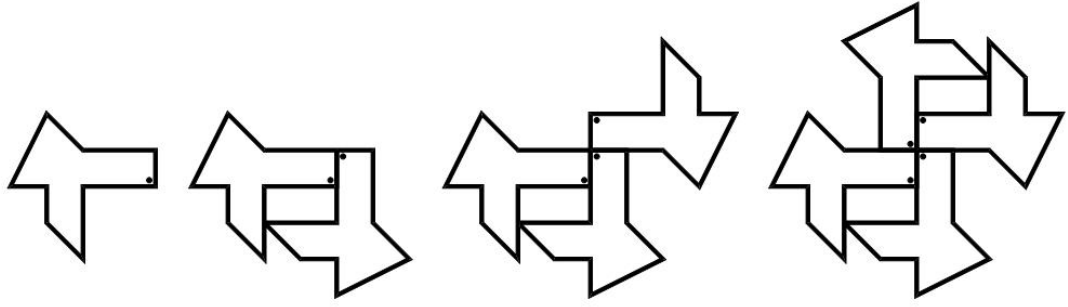
Şekil 2.7 Biçimler arası ilişkiler

Biçimler türetilirken bir başlangıç biçimine; öteleme, döndürme, yansıtma, ölçekleme, ekleme, çıkarma gibi aritmetik biçim operasyonları uygulanır.

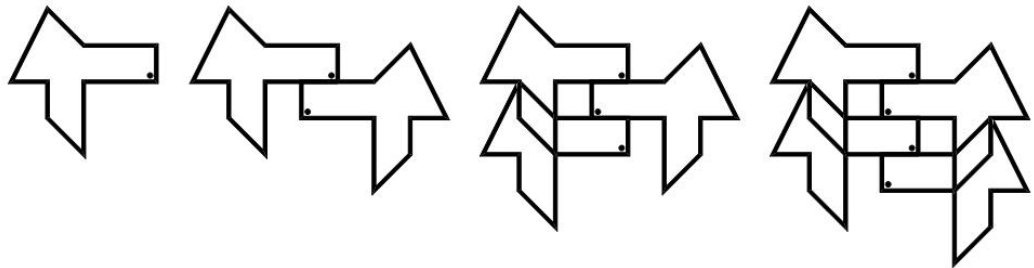
Bir başlangıç biçiminin türetilmesiyle oluşan, türeyen ve başlangıç biçimi arasında tanımlanan türeme ilişkisi sayısı, oluşacak sonuç ürün çeşitliliğini artırır. Aşağıda A1 ve A2 arasındaki ilişkinin kuralının değiştirilmesiyle oluşan formlar görülmektedir. Biçimler etiketlenilerek tanımlanan farklı ilişkilerin takibi kolaylaştırılmaktadır.



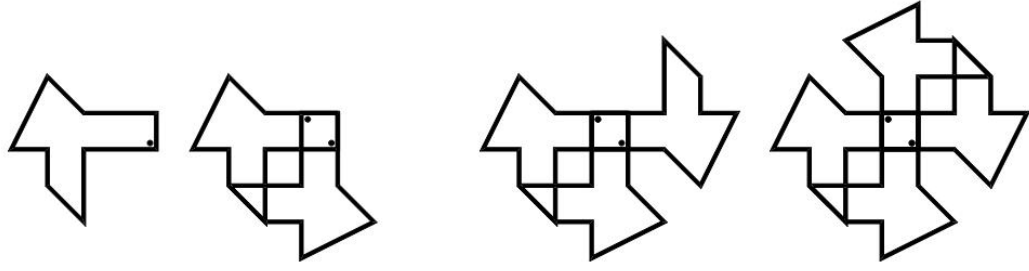
Şekil 2.8 A1 ile A2 arasındaki ilişki-1



Şekil 2.9 A1 ile A2 arasındaki ilişki-2



Şekil 2.10 A1 ile A2 arasındaki ilişki-3



Şekil 2.11 A1 ile A2 arasındaki ilişki-4

2.2.1. Karmaşık Gramerler

Karmaşık gramerlerde birden fazla biçimin birbirleriyle olan ilişkileri ve kuralları tanımlanır. Temel biçim gramerinde tanımlanmış kurallarla oluşan biçim kompozisyonları ve kural setleri başka bir kural setinin içinde yer alıp yeni bir kompozisyon meydana getirebilir. Biçim gramerlerine uygulanan aritmetik biçim operasyonları biçimlere uygulandığı gibi bu biçimlerin oluşturulduğu kompozisyonlarla türetilebilecek karmaşık gramerlere de uygulanırlar. Karmaşık gramerlerle tasarım sürecinde bilgisayar destekli teknikler kullanıldığında farklı biçimler arası farklı ilişkilerle olasılıklar uzayı öngörülemediği kadar artmaktadır.

Aşağıda Şekil 3.7’de Çolakoğlu tarafından yürütülen çalışma kapsamında Konur Gönenç’in yaptığı tasarım süreci karmaşık gramerler kullanılarak üretilen konutun perspektifi bulunmaktadır. Çalışmada farklı biçim ve farklı ilişkiler kullanılmıştır.

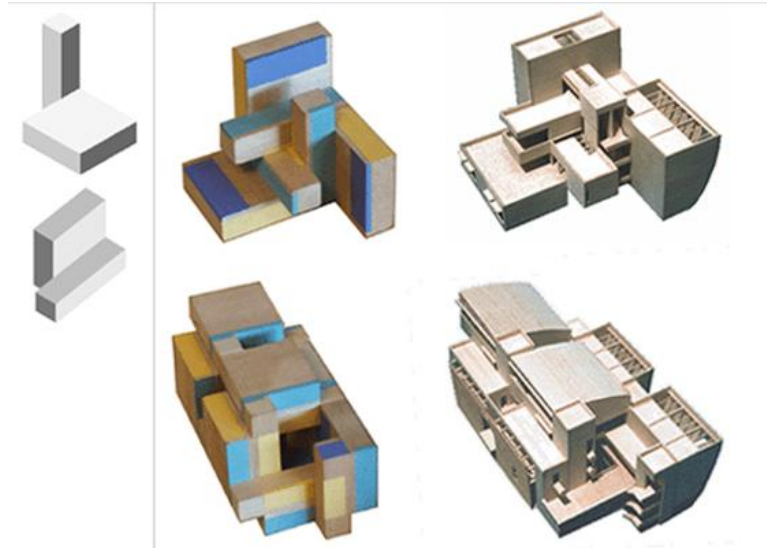


Şekil 2.12 Konut (Konur Gönenç, YTÜ)

2.2.2. Renk Gramerleri

Renk gramerleri analiz çalışmalarında ve üretici gramer tekniği olarak kullanılmaktadır. Renk gramerlerinde renklere yüklenecek fonksiyonlara bağlı olarak standart ve karmaşık biçim gramerlerindeki gibi olasılıklar uzayı daha da artacaktır. Başlangıç biçimine ve kurallara renk eklenir. Renkli yüzeyler veya kütlelerle temel veya karmaşık biçim gramerleri kuralları uygulanarak türetmeler yapılır. Biçim gramerlerinde ve karmaşık gramerlerde kullanılan aritmetik biçim operasyonları renk gramerlerinde de kullanılır.

Aşağıda Knight [5] arşivinde yer alan Jin Ho Park'ın tasarladığı müze projesinin gelişimi incelendiğinde renk gramerlerinin kullanıldığı görülmektedir. Bazı yüzeylerin tamamına bazı yüzeylerin de bir kısmına renk verilerek türetmeler yapılmıştır. Verilen renkler farklı ilişkileri ve işlevleri tanımlamaktadır.



Şekil 2.13 Müze (Jin Ho Park, MIT)

2.2.3. Froebel Blokları

Frederick Froebel 'in anaokulu çocukların geometrik bilgisine ve soyut düşünme becerisine katkıda bulunmak üzere eğitim sürecinde kullanılması için tasarladığı ahşap bloklardır.



Şekil 2.14 Froebel Blokları

Froebel blokları ile oynayarak büyüyen Frank Lloyd Wright'ın yaptığı binalara bakıldığında bu etki hissedilmektedir. Bunlardan en ünlüsü Şelale Evi'dir.

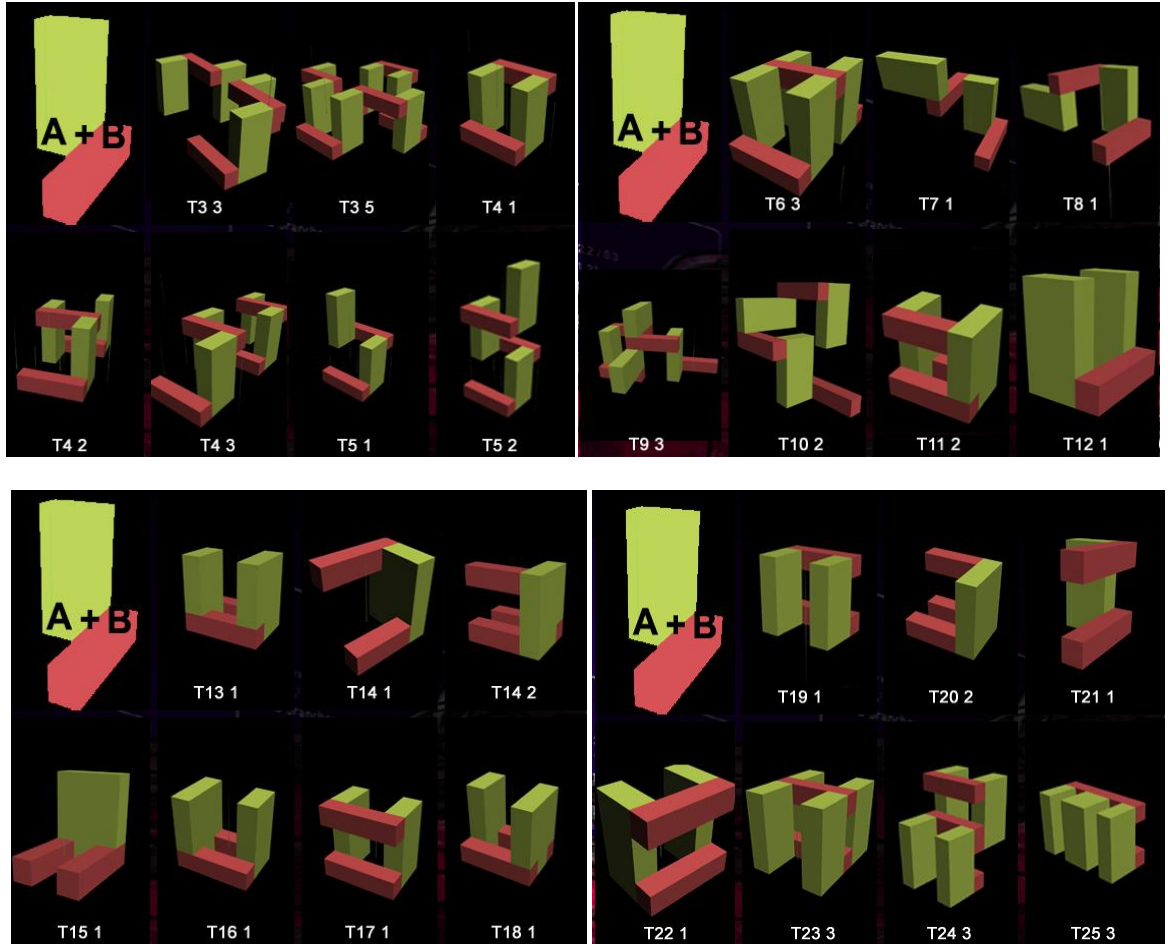


Şekil 2.15 Şelale Evi (Frank Lloyd Wright)

Froebel blokları üç boyutlu biçimlerden oluşmaktadır ve öteleme döndürme simetri alma gibi aritmetik biçim operasyonları kullanılmaktadır. Blokların üç boyutlu olması, el ile tasarım sürecinde kullanımını kolaylaştırmaktadır. Blokların birbirileri ile olan ilişkileri tanımlanırken, iki blok arasındaki ilişkinin takibi için etiket noktaları belirlenir ve türetilenler

yapılır. Blokların birbirleri ile olan ilişkileri değıştikçe türetilen üründe değışecektir. Böylece aynı bloklardan oluşan, aynı dili konuşan bir tasarım uzayı elde edilecek ve tasarım süreci tasarımcının bu formları yorumlaması şeklinde devam edecektir.

Şekil 3.11’de froebel blokları kullanılıp, A ile B arasındaki ilişki tanımlanmıştır. Maxscript ile aritmetik biçim operasyonları kullanılarak türetmeler yapılmıştır. T3 3 üçüncü ilişkinin üçüncü aşamasını, T3 5 beşinci aşamasını göstermektedir. Aynı şekilde T4 1 dördüncü ilişkinin ilk aşamasını, T4 2 ikinci aşamasını, T4 3 üçüncü aşamasını göstermektedir. Diğer seçeneklerde de çeşitli ilişkilerin bazı aşamaları gösterilmiştir.



Şekil 2. 16 Bloklarla script kullanarak form türetme

Türetmeler sonucunda ulaşılabilecek sonucun niteliği tasarımcıya bağlıdır. Tasarımcı script dilini iyi kullanabilmeli ve tasarım amacına uygun algoritmaları oluşturabilmelidir. Daha sonra türeyecek formları hayal gücü ve öngörüsü ile yönlendirecektir.

2.3. Biçim Gramerleri Uygulamaları

Biçim gramerleri tasarım sürecinde analiz yöntemi ile ve üretici tasarım tekniği olarak kullanılmaktadır. Analiz çalışmalarında, incelenen formların biçimsel ve paralel çalışan tanımlayıcı gramerleri irdelenir. Elde edilen bu gramerler hedeflenen program paralelinde yeni kural setleri eklenerek veya mevcut kurallar değiştirilerek yeni ürünlerin elde edilmesine yönelik kullanılır. Aynı şekilde bir tasarımcının tasarımları incelenerek, kendi stili ve örüntü dili analiz edilebilir ve bu analiz sonunda elde edilen gramerler ile o tasarımcının mimari dilinde tasarımlar üretilebilir. Biçim gramerlerinin analiz yöntemiyle Çin buz desenleri, kilim motifleri ve İslami süslemeler üzerine yapılan çalışmalarda; biçimler arası ilişkiler irdelenerek gramerleri elde edilmiş ve elde edilen gramerlerle aynı uygulama tekniği ile farklı varyasyonlarla benzer nitelikte ürünler elde edilmiştir.

Biçim gramerlerinin üretici tasarım tekniği olarak kullanılmasına ilişkin örnekler ise, bilgisayar yazılımlarıyla paralel bir şekilde gelişim göstermiştir. Yazılımların yeterli ve iyi kullanılıyor olması, sonuç ürün zenginliğini artırmıştır.

Analiz Çalışmaları

İlk dönem biçim grameri çalışmaları analiz ağırlıklıdır. Bu çalışmalar, biçim grameri kuramının tasarım kuramı, bilgisayar destekli tasarım (CAD) ve ilişkili alanlarda kabul gören bir paradigma olmasını sağlamıştır [4]. Biçim gramerleri ile ilgili olarak ilk çalışma Stiny tarafından 1977 de yazılan“ Ice-Ray: a note on the generation of Chinese lattice designs” isimli makaledir. Bu çalışmada Çin buz deseni kompozisyonları beş basit kural içeren gramer ile anlatılmış, gramer kuralları kullanılarak mevcut Çin buz desenlerinin ve aynı stilde sayısız tasarımın yapılabileceği gösterilmiştir. Buna benzer çalışmalar İslam süsleme motiflerinin ve Selçuklu halı desen gramerlerinin çıkarılmasında uygulanmıştır.

Bu çalışmayı Stiny ve Mitchell’ in ürettiği Palladian mimarı grameri izlemiştir. Mimarlık alanındaki gramer tabanlı diğer çalışmalara örnek olarak; Guiseppe Terragni, Frank Lyod Wrigt, Glenn Murgutt, Christopher Wrenn, mimari sitil analizlerine örnek olarak; Japanese tea rooms, bungalows of Buffalo, Queen Anne Houses ve peyzaj mimarisi analiz örneği olarak da Mughul bahçeleri gösterilebilir [4].

Sentez Çalışmaları

İkinci dönem biçim grameri çalışmaları Frederic Froebel' in çocuk yuvası yöntemi ile mimari tasarım atölyesinde uygulanan tasarım yöntemlerinin benzerliğinden yola çıkarak geliştirilen üç boyutlu tasarım kompozisyonları üretme çalışmalarıdır. Alman eğitimci Froebel tarafından 19. yy.ın başında geliştirilen yöntemde üç farklı form tanımlanmıştır. Bilim formları ile çocukları, sayılar ve oran ilişkilerinde görülen soyut mantık ilişkileri ile tanıştırmaktadır. Sanat formları ile blokların simetri kurallarına göre bir araya getirilmesi sonucu oluşan soyut mekânsal ilişkileri algılamalarını geliştirmeyi amaçlamaktadır. Doğa formları ile ise gerçek hayatta objeler arasında gözlemediğimiz ilişkileri analiz etmelerini hedeflemektedir. Bu yöntem ile çocuk, oyun içerisinde düşünce organizasyonunu sağlamayı, gözleme dayalı seçim yapabilmeyi ve kuralları farkında olarak uygulamayı öğrenmektedir [4].

Stiny “Kindergarden grammars: designing with Frobel’ building gifts” makalesinde Froebel in çalışmasını referans alarak geliştirdiği “Froebel Blok Grammar” yapıcı tasarım yöntemini tasarım atölyelerinde uygulamayı önermiştir. (Stiny1980) Bu yöntem UCLA ve MIT de Knight[5], Yıldız Teknik Üniversitesinde, Çolakoğlu tarafından lisans ve master düzeyindeki tasarım atölyelerinde tasarım sürecinin farklı aşamalarında uygulamıştır [4].

Hibrit Çalışmalar (Analiz + Sentez)

Üçüncü dönem biçim grameri çalışmaları analiz ve sentezin birlikte kullanıldığı orijinal yeni tasarımlar üretmeye yöneliktir. Bu yöntem ile bir tasarım dili, önce o dili oluşturan gramer kuralları çıkarılarak analiz edilir, daha sonra bu kurallar kullanılarak ya da değişiklikler yapılarak yeni tasarım dili oluşturulur ve yeni ürünler elde edilir.

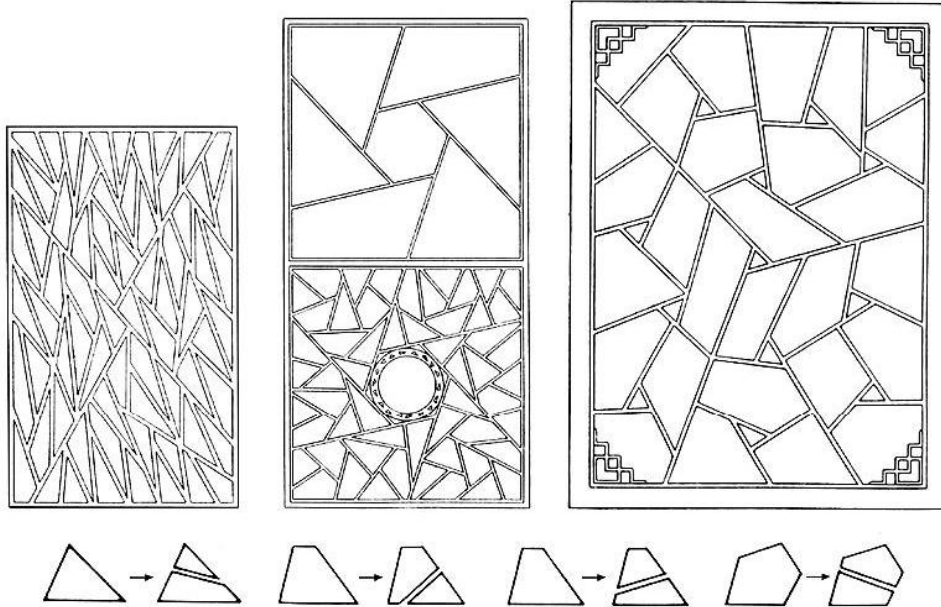
Analiz-Transformasyon-Sentez süreçlerinden oluşan bu yöntem MIT, Harvard, Yale üniversitelerinde Julia Eizenberg, CMU da Flemming UCLA ve MIT de Terry Knight, Yıldız Teknik Üniversitesinde Birgül Çolakoğlu tarafından mimari atölye eğitiminde kullanılmıştır [4]. Atölyelerde öğrenciler bir mimarın veya belirli bir stilin, dönemin mimarisini analiz ederek tasarım kurallarını çıkarmış daha sonra bu kurallar ile oynayarak verilen programdaki ihtiyaçları karşılayan kendi tasarım kurallarını oluşturmuşlardır. YTÜ Mimarlık Fakültesi Bilgisayar Ortamında Tasarım Yüksek Lisans Programında yürütülen Gramer Tabanlı Tasarım Atölyesinde, “Mardin de Yeni Konut Tasarımı” projesi bu yöntemle geliştirilmiştir [4].

2.3.1. İki Boyutlu Biçim Gramerleri Uygulamaları

2.3.1.1. El ile Yapılan İki Boyutlu Biçim Gramerleri Uygulamaları

Ice-ray grammar (Stiny, G.)

Çin'deki iceray pencere kafesi olarak bilinen cam üzerine uygulanan buz desenlerinin gramerlerinin çıkarılmasının hedeflendiği G.Stiny tarafından yapılan çalışmada Daniel Sheets Dye tarafından oluşturulan kataloglar incelenmektedir. Bu kataloglarda M.Ö. 1000 ile M.S. 1900 yılları arasında yapılan geleneksel Çin pencere kafeslerinin örnekler bulunmaktadır. Bu katalogda yer alan örneklerin sahip oldukları periyodik düzenli yapıları basit biçim gramerlerinin üretilmesine olanak sağlar ama Çin pencere kafesleri bu denli düzene ve periyodikliğe sahip değildir (Stiny 1977).

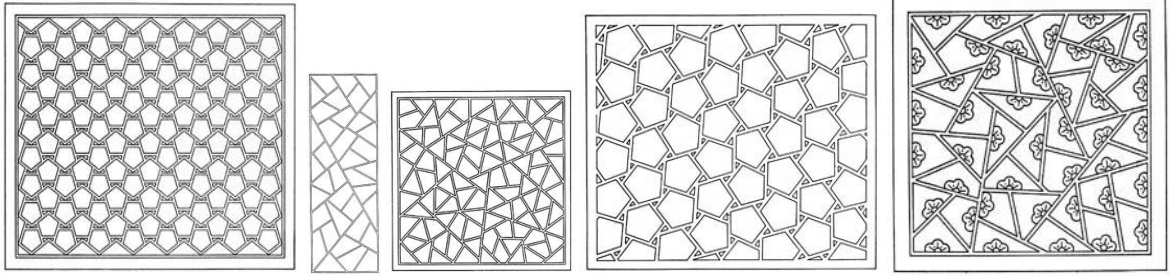


Şekil 2.17 Buz ışını analizleri

G. Stiny'e göre bu tip bir tasarımı değerlendirebilmek, soğuk bir gecede durgun bir su üzerindeki buz formlarının görmekle mümkün olabilecektir. G. Stiny'e göre düz çizgiler ve uzun çizgiler güzel ve benzersiz bir model oluşturmak için birleşirler. Çinliler buna "ice line" diyorlar ve bu da buzdaki çatlaklardan oluşan çizgiler anlamında kullanılmaktadır (Stiny 1977).

Tanımladıkları parametrik biçim gramerleri ile basit bir şekilde ve sezgisel olarak işleyen bir yöntemle ice-ray'ı üretirler. Çalışmanın sonunda getirdikleri bir buz deseni zanaatçısından yapmasını istedikleri uygulamada zanaatçının uygulama esnasında makalede tanımlanan kuralları uyguladığı görülmüştür (Stiny 1977).

Stiny yaptıkları çalışma ile Çin buz desenlerini 4 ana kategoride sınıflandırmıştır.



Şekil 2.18 -1: Parametrik Biçim Gramerleri (Stiny 1977)

Şekil 3.13'teki resimlerden birincisi "Parametrik Biçim Gramerleri", ikinci ve üçüncüsü "Parametrik Gridal Biçim Gramerleri", dördüncüsü "Sınırlı Parametrik Biçim Gramerleri", ve beşincisi ise "Bilinmeyen Parametrik Biçim Gramerleri"dir (Stiny 1977).

The palladian grammar (Stiny, G., Mitchell, W-J)

Stiny ve Mitchell, Palladio villalarının zemin kat planlarının gramerlerinin çıkarılması ve bu gramerin Palladian stilin bir tanımı olarak geliştirilip yeni villaların zemin kat planlarının türetilmesini hedeflemişlerdir (Stiny, Mitchell 1978).

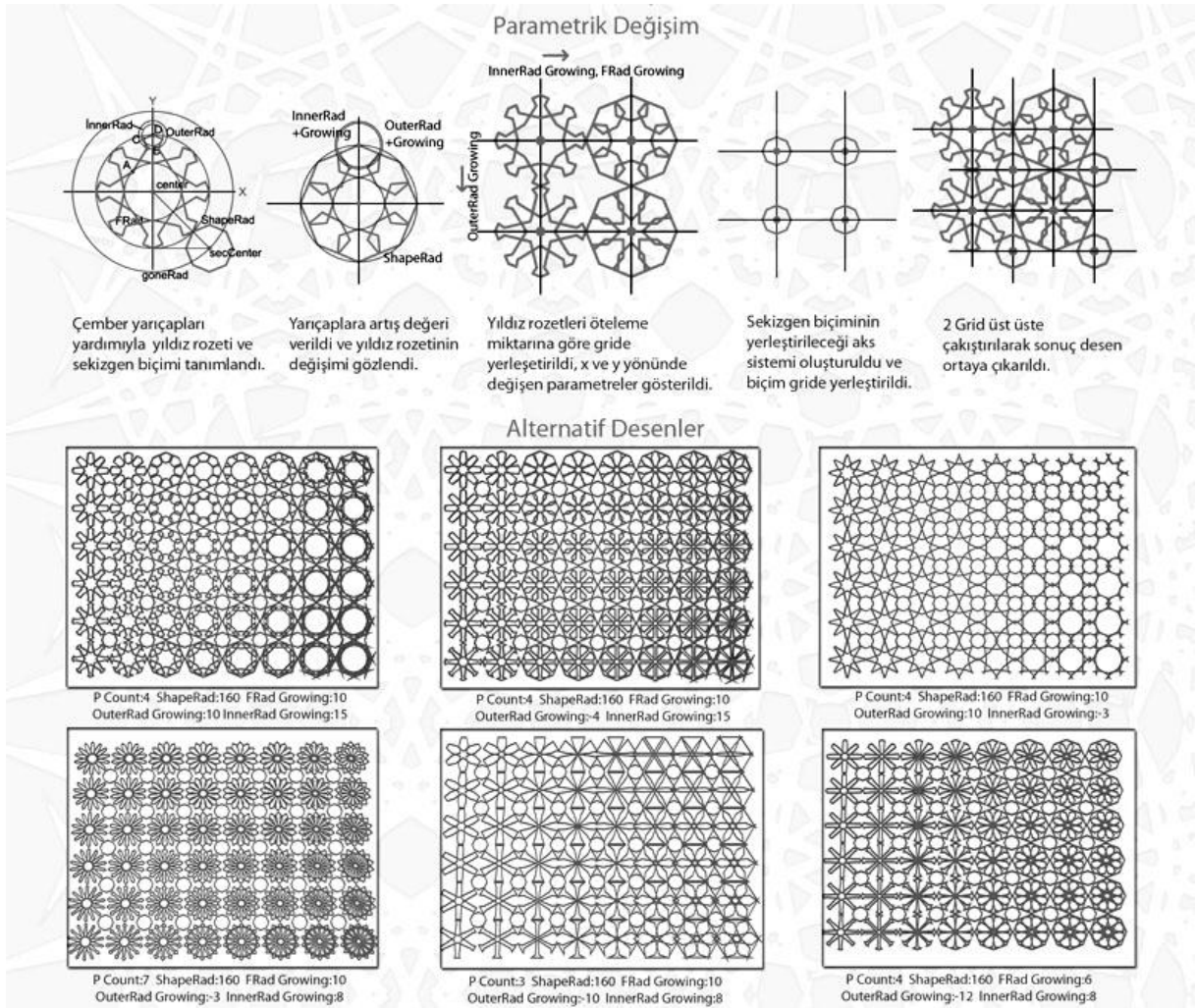
Öncelikle Palladio'nun mimari yapısı grameri ve parçaları yeniden incelenmiş daha sonrada zemin kat planlarını üretecek parametrik biçim gramer kuralları belirlenmiştir. Villa planları Palladio sitilini temsil ettiği için dekoratif süslemeler gibi uygulamaların araştırılması yerine Palladio villalarının planlarının geometrik özellikleri incelenmiştir. Quattro Libre'deki Palladio Villa planlarında tek ve çift akslı olmak üzere bir sınıflandırmaya gidilmiştir. Buna göre tek akslı olan villa tiplerinin üretilmesi için izlenilecek sekiz aşamayı belirlemişlerdir [6].

1: Aksların tanımlanması 2: Dış duvarların tanımlanması 3: Odaların yerleşimi 4: İç duvarların yeniden yerleştirilmesi 5: Girişler ve dış duvar yükselteleri 6: Dekoratif öğeler, kolonlar 7: Pencere ve kapılar 8: Bitişler

The Villa Malcontenta'nın planı yukarıda yazılı süreçler uygulanarak elde edilmiştir. Ayrıca çift akslı villa plan tiplerinin üretilmesi için tek akslı villa tiplerinin üretim aşamalarına eklenmesi gereken adımlar belirlenmiş buna göre uygulama yapılmış. Çift aks yapısına göre tasarlanan villa tiplerinden en bilineni ise Villa Rotonda'dır.

2.3.1.2. Dijital İki Boyutlu Biçim Gramerleri Uygulamaları

Üç boyutlu tasarım programlarına eklenen script dilleri ile kullanıcı arayüze müdahale etme şansı bulmakta ve istediği uygulamayı gerçekleştirebilecek algoritmaları geliştirebilmektedir. Bu doğrultuda Yıldız Teknik Üniversitesi Bilgisayar Ortamında Tasarım Yüksek Lisans programında Birgül Çolakoğlu tarafından verilen Proje 1 [7] dersi kapsamında yapılan atölye çalışmasında script dilleri öğrenciye öğretilmiş ve uygulama çalışması tasarım sürecinde kullanılması beklenmiştir. Algoritmik düşünce mantığının ve parametre kavramının da verildiği ders sürecinde öğrenciler, yaptıkları analizlere göre geliştirdikleri algoritmaları üç boyutlu programların script dillerini kullanarak iki boyutlu desenler, dokular türeten arayüzler geliştirmişlerdir. Yapılan türetmelerle elde edilen sonuçlara ilişkin bazı örnekler ve uygulamaları aşağıdaki resimlerde yer almaktadır.



Şekil 2.19 İki boyutlu biçim gramer uygulama örnekleri (Sevkan Harman Y.T.Ü)



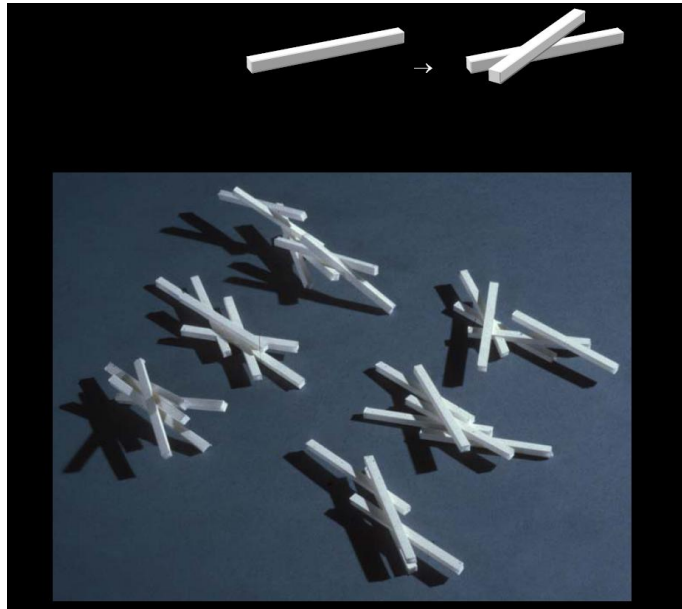
Şekil 2.20 İki boyutlu biçim gramer uygulamaları ve üretilmiş örnekler [7] (Y.T.Ü)

2.3.2. Üç Boyutlu Biçim Gramerleri Uygulamaları

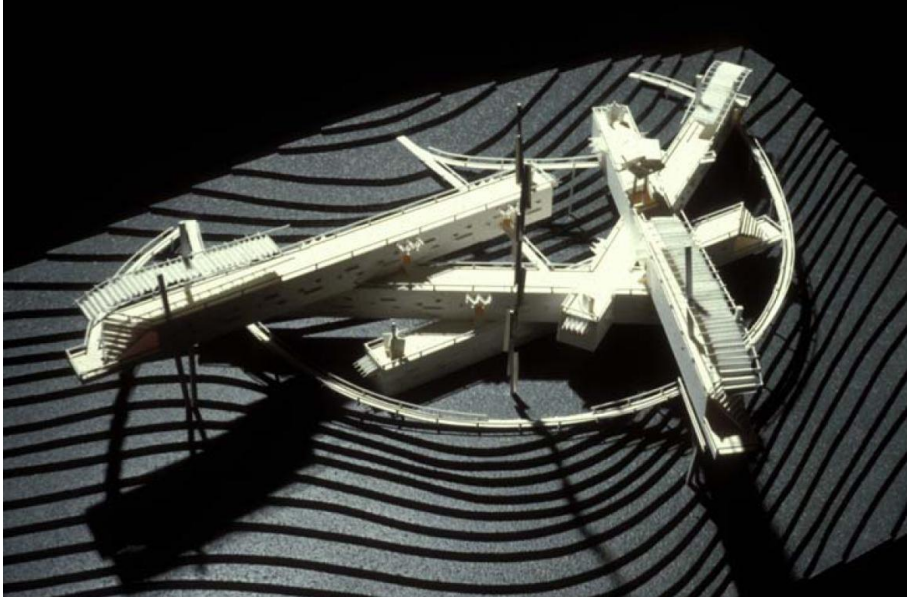
Üç boyutlu biçim gramerleri uygulamalarında froebel blokları kullanılarak yeni ve özgün tasarımlar üretmeye yönelik form bulma çalışmaları yapılmıştır. Ayrıca üç boyutlu biçim gramerleri ile mevcut bir tasarım dilinin yönteminin analizi ve sentezi ile tasarım yapmayı hedefleyen hibrit çalışmalar da yapılmıştır.

2.3.2.1. El ile Yapılan Üç Boyutlu Biçim Gramerleri Uygulamaları

Historical museum, San Gimignano, Italy (Randy Brown)

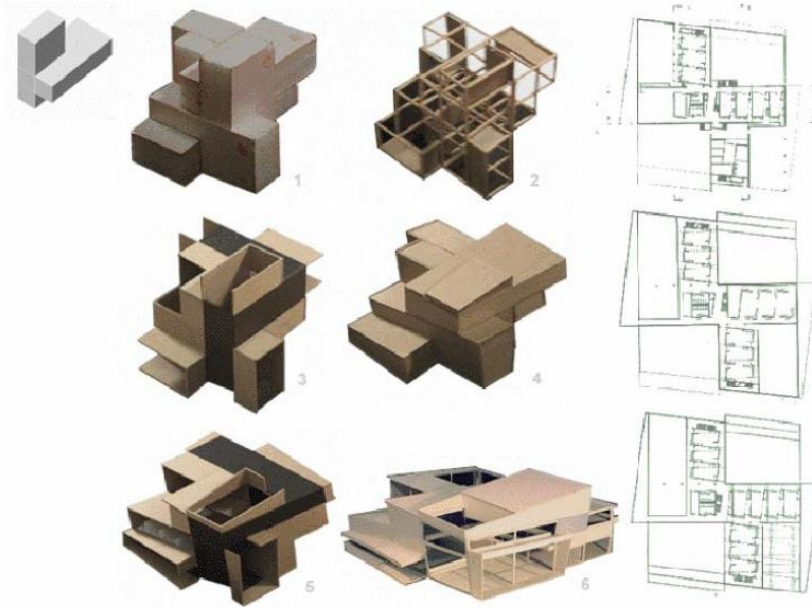


Şekil 2.21 MIT Terry Knight atölyesi [5] 'Tarih Müzesi' Kurallar ve türetmeler (Randy Brown, MIT)

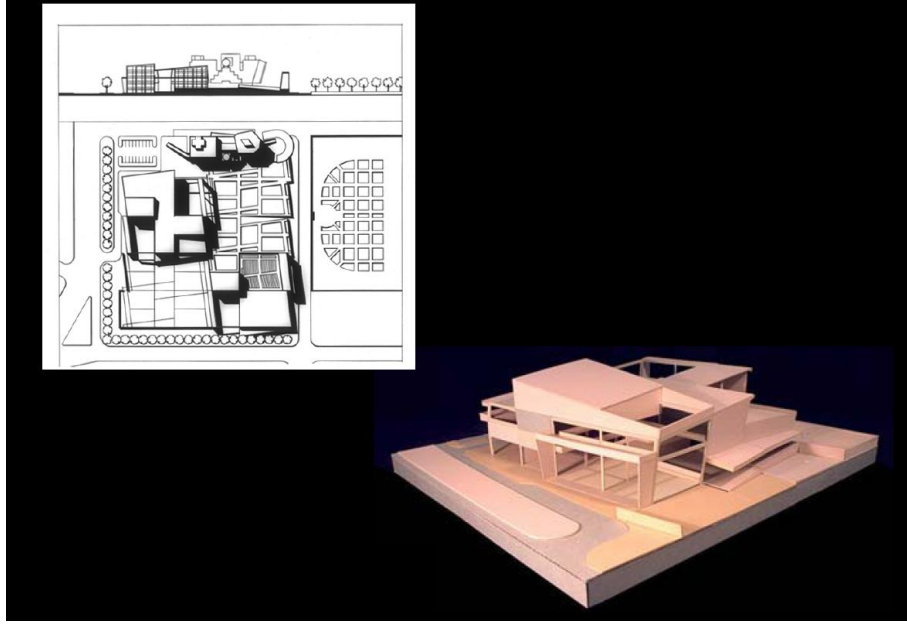


Şekil 2.22 MIT Terry Knight atölyesi [5] ‘Tarih Müzesi’ maketi (Randy Brown, MIT)

Elementary School Complex, Los Angeles, (Michael Brown)



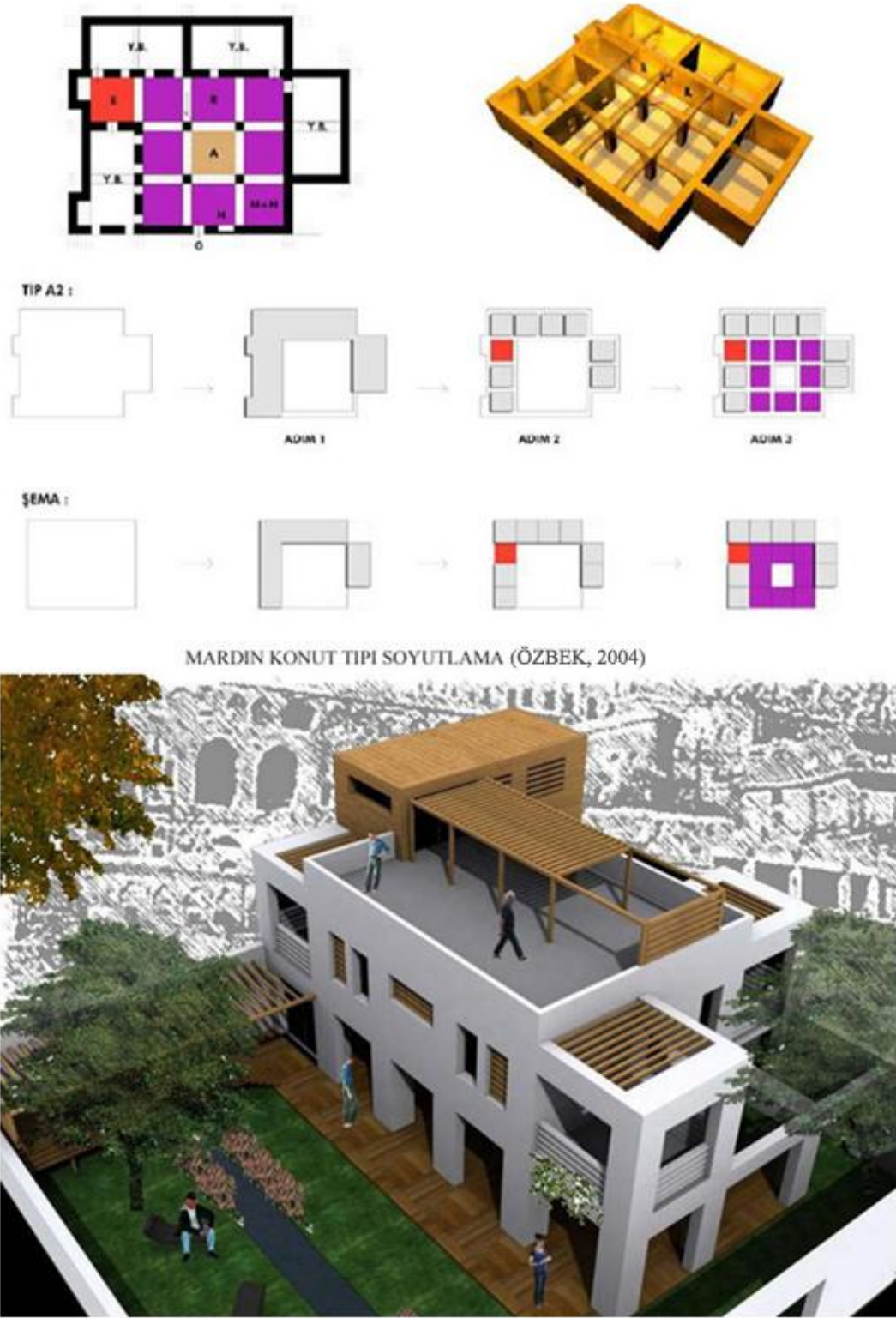
Şekil 2.23 MIT Terry Knight atölyesi [5] ‘Elementary School Complex’ Tasarım süreci
(Michael Brown, MIT)



Şekil 2.24 MIT Terry Knight atölyesi 'Elementary School Complex' Maket ve vaziyet
(Michael Brown, MIT)

Mardin Projesi, (Togay Özkaraduman)

YTÜ 'de Birgül Çolakoğlu tarafından yürütülen Mardin projesi kapsamında yapılan yüksek lisans tez [8] çalışmasında Mardin konut grameri kuralları tasarıma başlama noktası olarak kullanılmıştır. Bu kuralları inceleyerek tasarım sürecinde iklim, mekan ilişkileri, malzeme, mekan kullanışlarını gibi etmenler dikkate alınarak yeni Mardin konutları için yeni tasarım kuralları oluşturulmuştur. Çalışma Özkaraduman tarafından el ile geliştirilmiş, dijital ortamda modelleme ve sunuşu yapılmıştır.

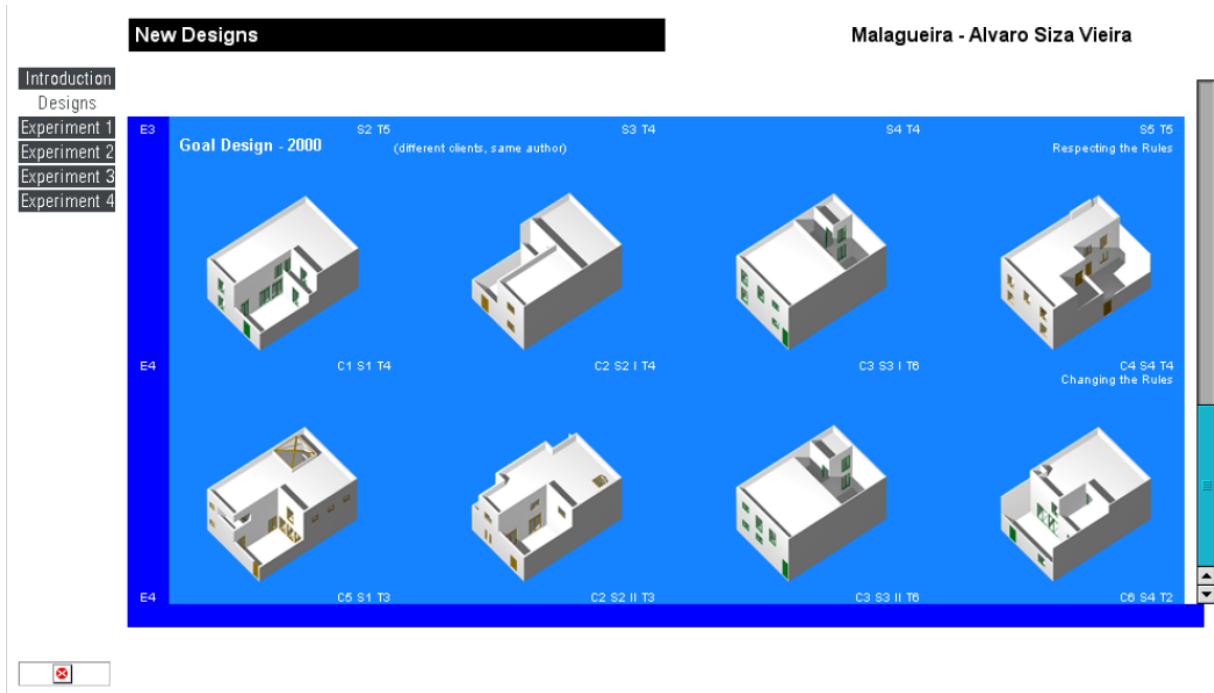


Şekil 2.25 Gramer tabanlı konut tasarımı (Togay Özkaraduman, YTÜ)

2.3.2.2. Dijital 3 Boyutlu Türetici Gramer Uygulamaları

Alvaro Siza Vieira Malagueira Evleri

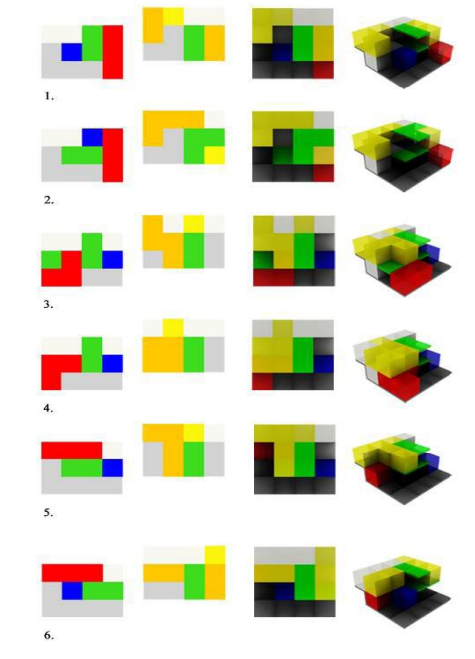
Malagueira Evlerini Portekiz’de Evora yakınlarında Alvaro Siza Vieira tasarlanmıştır. 8x12’lik alanlar üzerine tasarlanan, birden beşe kadar değişik sayıda yatak odasına sahip, toplam 1200 konutluk bir projedir. Siza’nın mimari tasarım yaklaşımındaki biçim gramerlerine uygunluk sayesinde Malagueira Evlerinin analizi kolay yapılabilmektedir. Doktora çalışması [9] olarak geliştirilen bu çalışmada bir biçim grameri yorumlayıcısı geliştirmiştir. Bu yorumlayıcı ile rastgele form türetmeleri yapılmamaktadır. Amacı belirlenen kurallar çerçevesinde bir konut tasarımı oluşturmaktır (Duarte, 2001).



Şekil 2.26 Yeni türetilen Malagueira Evleri (Duarte, 2001)

Mardin Projesi (Belinda Torus)

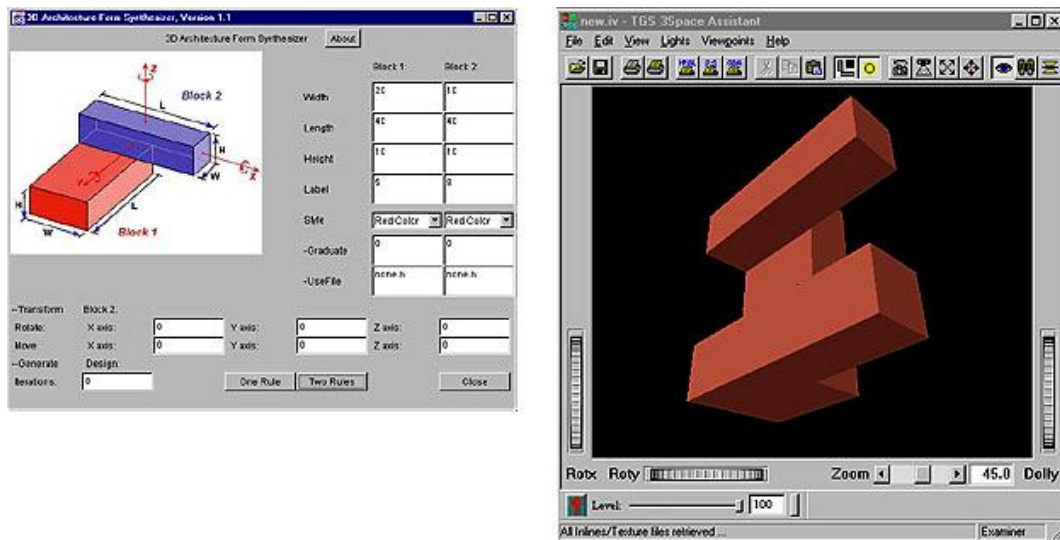
2005 yılında Yıldız Teknik Üniversitesi Bilgisayar Ortamında Mimarlık Yüksek Lisans Programı proje atölyesinde Birgül Çolakoğlu’nun yönettiği Mardin projesinde yapılan analizler ve çalışmalarla kurallar belirlenmiş ve ilk önce 3ds Max programında “script”ler yazılarak rastlantısal türetmeler yapılmıştır. Daha sonra kent, çevre dokusuna ve tasarımcı sezgilerine göre yorumlanmıştır (Torus, 2005). Bu çalışmanın da amacı belirlenen kurallar çerçevesinde yeni bir konut tasarımı oluşturmaktır.



Şekil 2.27 Tasarlanan program ile üretilen formlar (Belinda Torus, YTÜ)

3D Architecture Form Synthesizer (Yufei)

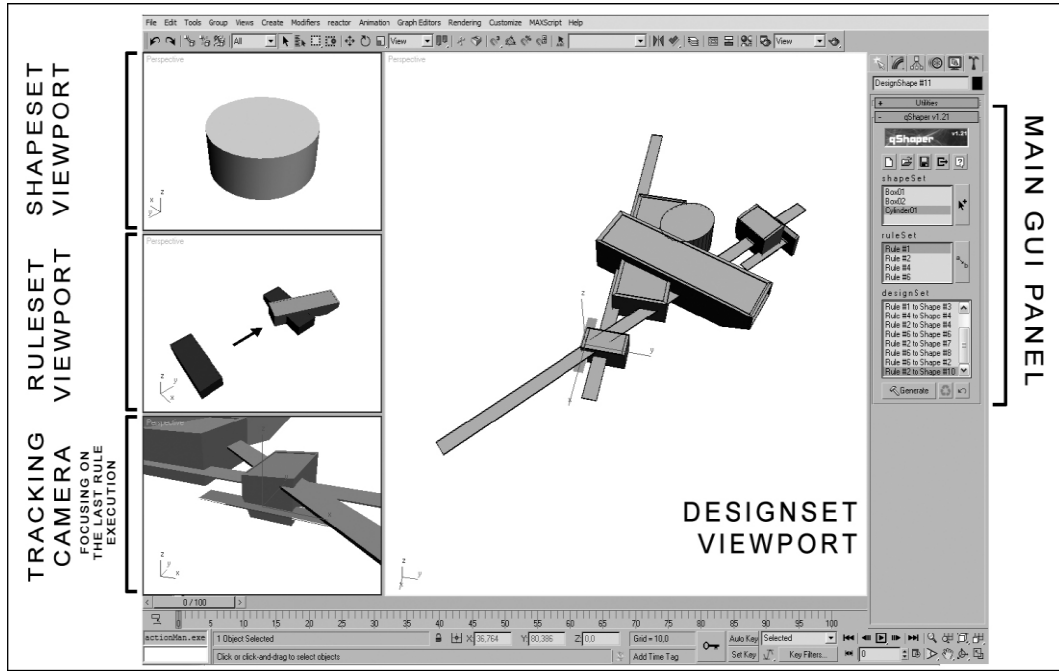
“3dshaper” olarak da tanımlanan arayüz “MIT-MIYAGI” adlı Gabriela Celani tarafından düzenlenen biçim gramerleri ve bilgisayar uygulamaları ile ilgili eğitici bir deney yapmayı hedefleyen bir çalışma atölyesi kapsamında Yufei tarafından geliştirilmiştir [10]. Aynı çalışma gurubunda, McGill tarafından “2dShaper” ve Celani tarafından “AutoGrammar” adlı türetici arayüzler geliştirilmiştir. 3dshaper, biçim gramer kuralları ile tanımlanan parametreler ile form üretmeyi amaçlanmaktadır.



Şekil 2.28 -3D Architecture Form Synthesizer (Yufei, MIT)

Quick Shaper (Tuğrul Yazar, Birgül Çolakoğlu)

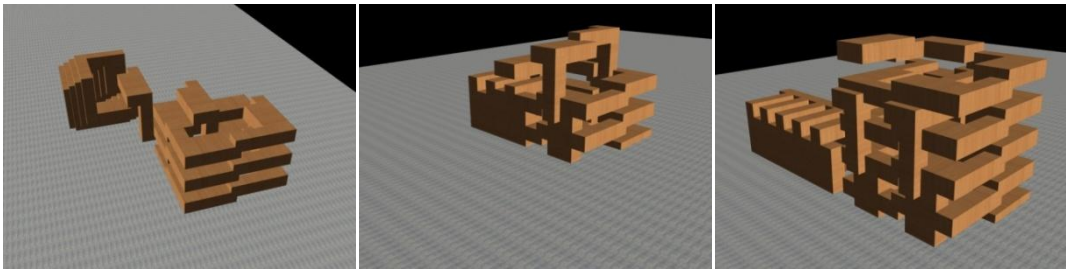
QShaper 3ds Max üzerinde maxscript kullanılarak geliştirilmiştir. Tasarımcılar veya öğrencilerin kural tabanlı tasarımlar üretmesini hedeflemektedir. Kullanıcıya kullanılan form seçiminde ve formların niteliklerine müdahalede esneklik sağlamakta, form parametrelerinin ve formlar arası ilişkilerin çek-bırak yöntemiyle belirlenebilmesi ile ise kullanımı kolaylaştırmaktadır.



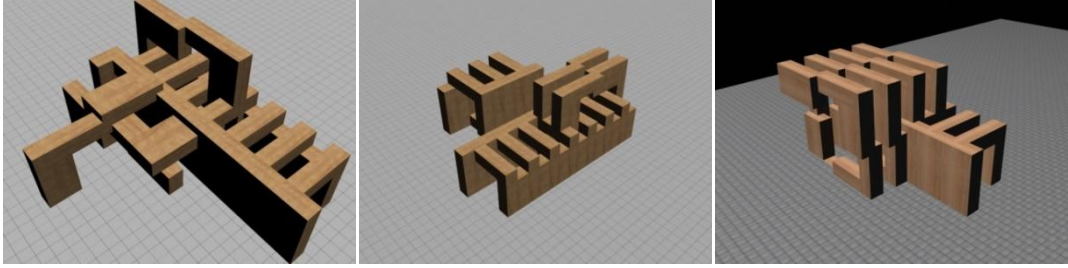
Şekil 2.29 QShaper (Tuğrul Yazar, Birgül Çolakoğlu YTÜ)

Konut Projesi (Gökhan Keskin, YTÜ)

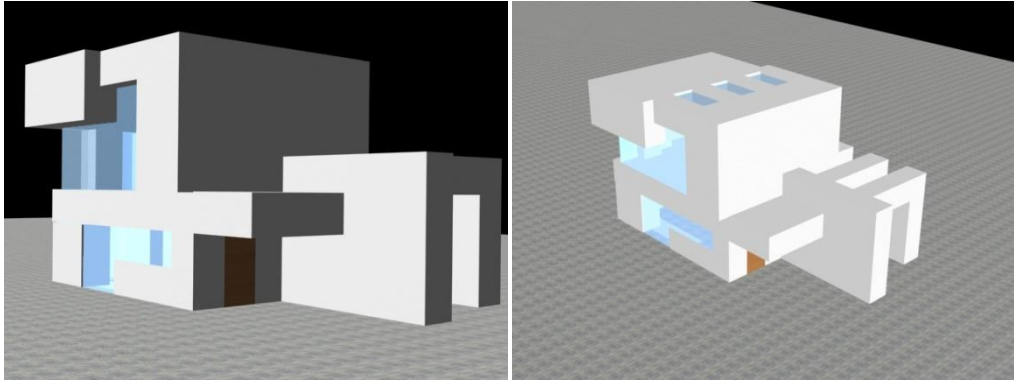
Y.T.Ü' de Birgül Çolakoğlu yürütücülüğündeki Biçim Gramerleri dersi kapsamında form üretmek için tarafımdan maxscript ile algoritmalar tasarlanmış ve script olarak yazılmıştır. 2 blok arası ilişkinin tanımlandığı script ile 3 adet döngü oluşturulmuş ve bu aşamalarda scripte müdahale edilerek bloklar arası ilişkiler değiştirilmiş, form üretme denemeleri yapılmıştır.



Şekil 2.30 Script ile türetmeler



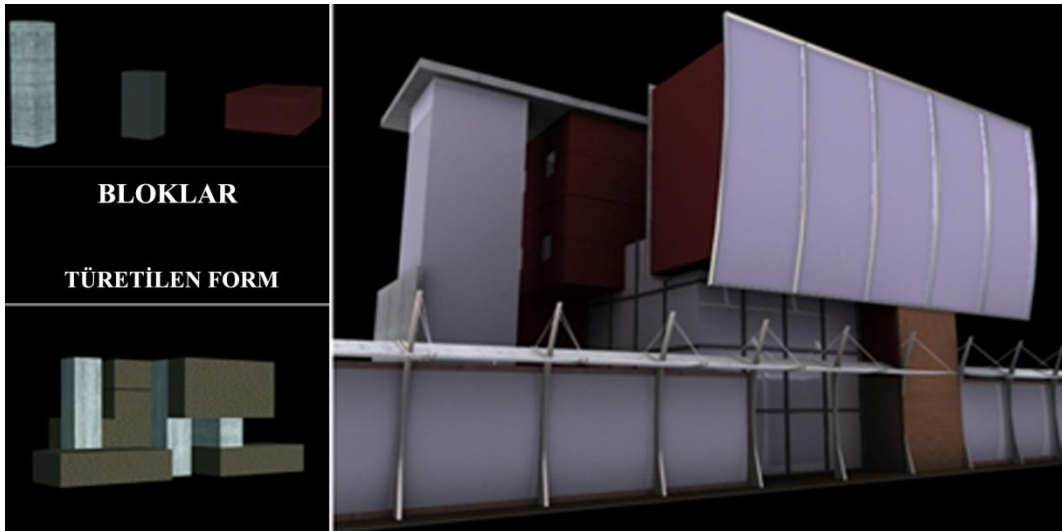
Şekil 2.31 Script ile türetmeler



Şekil 2.32 Yorumlama eskizleri

Kız Öğrenci Yurdu (Serkan Palabıyık, YTÜ)

Y.T.Ü' de Birgül Çolakoğlu yürütücülüğündeki Proje 3 dersi kapsamında Kız Öğrenci Yurdu olarak geliştirilen proje kapsamında form türetmek için maxscript kullanılmıştır. Elde edilen formlar arasından projeye en uygun olanı tasarımcı tarafından değerlendirilerek yorumlanmıştır. Aşağıdaki Şekil 2.33 ' de; bloklar, türetilen form ve formun yorumlanmış hali yer almaktadır.



Şekil 2.33 Kız Öğrenci Yurdu (Serkan Palabıyık, YTÜ)

3. NESNE TABANLI ALGORİTMİK TÜRETME MODELİ: FROEBEL FORM TÜRETİCİ

3.1. Nesne Tabanlı Türetme

Nesne Tabanlı Programlama

Nesne tabanlı programlamada yalnızca belirli bir veri yapısına ait veri türü tanımlanmaz Aynı zamanda bu veri yapısı belirli komutlara göre düzenlenmiştir. Bu durumdaki veri yapısından hem verileri hem de fonksiyonları içeren nesnelere ortaya çıkar.

Nesne tabanlı bir çizim programı çember, dikdörtgen veya üçgen gibi çeşitli türlerdeki farklı nesnelere tanıyabilir. Mesela yapılan işlem eğer üç boyutlu nesne oluşturmak ise bunun sonucunda bir küre, küp veya bir üçgen piramit elde edilir. Sonuç olarak, işlemin etkisi kullanıldığı nesneye bağlı olarak farklılık göstermektedir.

Nesne tabanlı programlamanın en büyük avantajı, programcıya modüller oluşturma imkân sunuyor olmasıdır. Bir kere oluşturulduktan sonra, başka bir nesne türü eklense dahi, modül içerisinde değişiklik yapmaya gerek kalmamaktadır. Bu sayede programcı, özelliklerinin çok büyük bir kısmını, hazır olan başka bir nesne içinden alan yeni bir nesneyi rahatlıkla oluşturabilmektedir. Bu nedenle nesne tabanlı programlamaları değiştirmek diğer programlara göre daha kolaydır. Nesne tabanlı programlamayı kullanabilmek için bu özelliği kullanan bir programlama dili gereklidir. Bu özellikteki en popüler programlama dilleri ise C++, Smalltalk ve Java'dır.

Nesne tabanlı bir sistem içerisinde nesnelere üst üste bindirilebilirler ve buna rağmen tek tek ulaşılabilir olarak saklanabilirler [11].

Nesne Tabanlı Türetme

Tasarımcıların nesne tabanlı programlama dillerini pratiği ve bilgisi olmadığı takdirde anlaması ve kullanması zor olmaktadır. Tasarımcılar mevcut yazılımların kısıtlı olmalarından dolayı kendi istedikleri işlevleri yükleyeceği programlar geliştirme gereği duymaktadırlar. Bu nedenle oluşan eksiklikleri gidermek için son dönemde özellikle CAD programlarına script dilleri eklenmektedir. Script dilleri makro dillerden olup sistem bütünleştiricilerdir. Script dilleri kullanıcıya, program içerisinde istediği işlevleri yerine getirecek arayüz oluşturma imkânı sağlamaktadır (MAYA, RHINO, GENERATIVE COMPONENTS,3dsMAX vb.).

3ds max, masaüstü sistemlerde en çok kullanılan 3 boyutlu modelleme, kaplama ve canlandırma yazılımıdır. Aynı zamanda, açık mimarisi sayesinde, üzerine en çok eklenti (Plug-In) geliştirilmiş yazılım olan 3ds max için ihtiyaç duyulan özel araçlara kolaylıkla ulaşabilmektedir. Birden fazla işlemciye sahip donanım sistemler için tasarlanmış yapı, gelişen sistem kaynaklarını en verimli şekilde kullanabilmektedir. Açık yazılım mimarisi, her işe yönelik eklentilerin (Plug-in) geliştirilmesine olanak sağlamaktadır. Character studio ve reactor bu alanda iki önemli örnektir. Önceden bir eklenti olarak satışa sunulan character studio, 3ds max 7'ye dahil edilerek yazılımın standart bir parçası haline gelmiştir. İçerdiği araçlar ile karakterlerin tüm hareketlerini en ince ayrıntısına kadar kontrol etmeye olanak sağlayan character studio 4, aynı zamanda "Crowds" sistemi ile kalabalık karakter gruplarının kendi içlerindeki etkileşimlerinin ve hareketlerinin kolayca canlandırılmasına olanak sağlamaktadır. Reactor, 3ds max için geliştirilmiş en kapsamlı fiziksel etkileşimli canlandırma sistemidir. Reactor, gerçek dünya fizik kurallarına uygun olarak yumuşak nesne dinamiği (soft-body dynamics), katı nesne dinamiği (rigid-body dynamics), kumaş, sıvı ve ip canlandırmalarını çözümlenmektedir. Bu sayede, Reactor ile hazırlanan canlandırmalarda nesnelere arası etkileşimin yanı sıra yerçekimi, rüzgâr ve kütle yoğunluğu gibi tanımlamalar da kullanılabilir.

Tüm araçların "Script"ler ile kontrol edilmesini sağlayan yapı, sıkça tekrarlanan işlemler için kullanıcıların kendi araçlarını geliştirmesini sağlamaktadır. Parametrik modelleme araçları ve değiştiriciler (Modifiers), yapılan modelleme çalışmalarının her aşamasında geçmişe yönelik düzenlemeler yapılmasını sağlar.

Diğer script dilleri gibi 3dsMaxscript ile tasarımcılar programlarda bulamadıkları özellikleri kendileri geliştirir ya da geliştirilmiş olanları alırlar. Scriptler ile her türlü özelleştirilmiş tasarım metodu geliştirilebilir. Yazılımları ağır dilleri düşünüldüğünde script dillerinin kavranması ve uygulanması da kolaydır. Script dilini ve o dille yapılan türetmelerin çıktılarının alınacağı programı iyi bilen tasarımcılar için geriye kalan istedikleri arayüzlerin algoritmalarını tasarlamaktır.

3.2. Tanım

Froebel form üretici 3dsMax tarafından programa eklentiler yapmak için sunulan Maxscript ile tarafımdan tasarlanmıştır.

Biçim gramerleri kurallarını kullanarak froebel blokları ile form üretmeyi sağlayan bir arayüzdür.

Form üretme belli analiz kurallarına göre değil tasarımcının öngörüsü ve sezileriyle rastgele yapılmaktadır.

Amaç, tasarımcının yorumlayabileceği nitelikte formlar üreterek yaratıcılığın ve üretkenliğin artırılmasını sağlamaktır.

Tasarlanan programda blok boyutları ve bloklar arası ilişkilerin girilerek kural setlerinin oluşturulduğu bütün parametreler kontrol edilebilmektedir.

Türetici 5 dönüşüm noktası ile türetmeler yapılacak şekilde tasarlanmıştır. 5 dönüşüm noktasının olmasının sebebi daha fazla kontrol noktasında arayüzün dahada uzayıp kontrolün karmaşıklaşması ve 5 dönüşüm noktasının form üretmek için yeterli görülmesidir.

3.3. Froebel Form Türetici

Form Türetici

Biçim gramerleri kurallarını kullanarak froebel blokları ile form türetmek için 3dsMax üzerinde bir arayüz ile çalışan(Şekil 3.1), maxscript ile geliştirilmiş bir eklenti(Plug-in)dir.



Şekil 3.1 Froebel form türetici arayüzü

Froebel form türetici arayüzünde kullanıcıyı yönlendiren parametreleri içeren 5 farklı seçenek bulunmaktadır.

1: Dönüşüm Noktaları 2: Kural Dizileri 3: Döngü Sayısı 4: Blok Boyutları 5: Blok İlişkileri

Arayüz; seçenekler, blok boyutları, blok ilişkileri ve dönüşüm noktaları ile ilgili kısımlardan oluşmaktadır.

Seçenekler

Dönüşüm Noktaları, Kural setleri, döngü sayısı gibi seçenekleri ve arayüzü çalıştıran butonu içerir.

Blok Boyutları

A1,A2,A3,A4 diye adlandırılan, form türetmede kullanacağımız blokların en boy ve yükseklik ölçülerini belirleyeceğimiz kutuları içerir.

Blok İlişkileri

Bloklar arasındaki ilişkileri biçim gramerleri kurallarına göre tanımlarız.

Dönüşüm Noktaları

Döngü sırasında belirli bir noktadan diğer bir noktaya kadar blok boyutlarını ve blok ilişkilerini yeniden tanımlamamızı sağlar.

Kural dizisi seçeneği ile kullanmak istediğimiz blok sayısını belirleriz. Kural dizileri ve blok sayısı ilişkileri aşağıdaki gibidir.

A1A2A1 ile 2 adet blok kullanılırız. Kural dizisi A1A2A1A2A1A2A1 şeklinde devam eder. Bu durumda A1 ile A2 ve A2'nin tekrar A1 ile olan ilişkisi tanımlanır. Döngü sayısını 3 olarak belirlersek oluşacak blok seti A1A2A1A2A1A2A1 olacaktır.

A1A2A3A1 ile 3 adet blok kullanılırız. Kural dizisi A1A2A3A1A2A3A1 şeklinde devam eder. Bu durumda A1 ile A2, A2 ile A3, ve A3'ün tekrar A1 ile olan ilişkisi tanımlanır. Döngü sayısını 3 olarak belirlersek oluşacak blok seti A1A2A3A1A2A3A1A2A3A1 olacaktır.

A1A2A3A4A1 ile 4 adet blok kullanılırız. Kural dizisi A1A2A3A4A1A2A3A4A1 şeklinde devam eder. Bu durumda A1 ile A2, A2 ile A3,A3 ile A4 ve A4'ün tekrar A1 ile olan ilişkisi tanımlanır. Döngü sayısını 3 olarak belirlersek oluşacak blok seti A1A2A3A4A1A2A3A4A1A2A3A4A1 olacaktır.

Blok Boyutları ile blokların X,Y,Z diye adlandırılan en boy ve yükseklik boyutlarını kutucuklara gireriz.

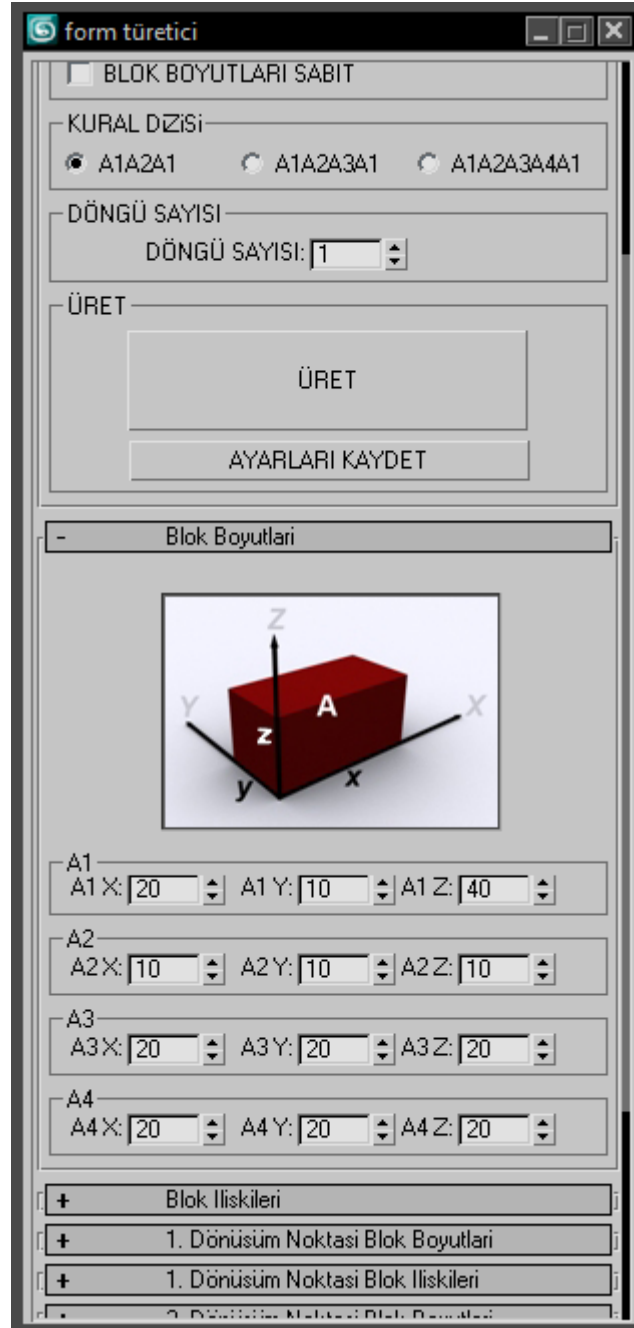
A1X, A1Y, A1Z; A1' in en boy ve yükseklik ölçüleridir.

A2X, A2Y, A2Z; A2' in en boy ve yükseklik ölçüleridir.

A3X, A3Y, A3Z; A3' in en boy ve yükseklik ölçüleridir.

A4X, A4Y, A4Z; A4' in en boy ve yükseklik ölçüleridir.

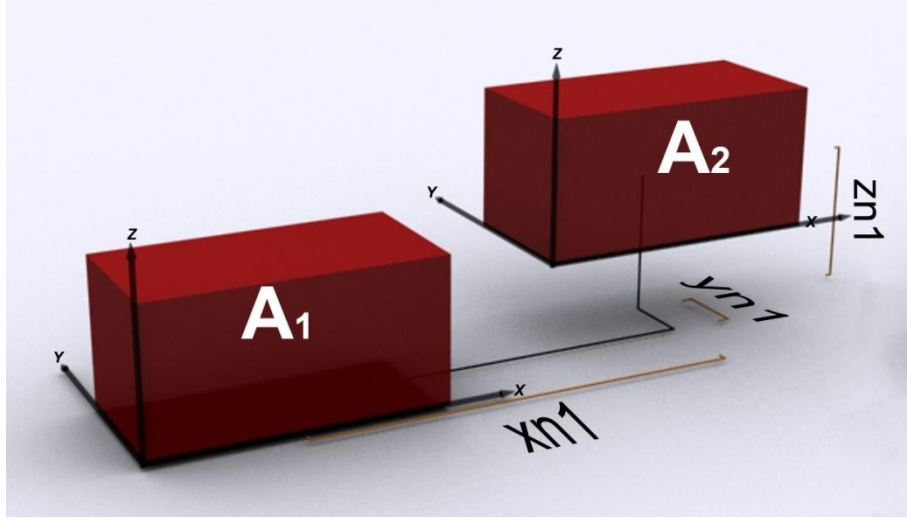
Blok boyutları 5 dönüşüm noktasında yeniden tanımlanabilir blokların ölçüleri değiştirilebilir.



Şekil 3.2 Froebel form üretici ile tasarlama

Blok İlişkileri:

Blok ilişkileri ile blokların birbirilerine göre konumlarını öteleme ve döndürme transformasyonları ile tanımlarız. Şekil 3.3'de A1 ile A2 arasındaki ilişki tanımlanmaktadır.



Şekil 3.3 A1 ile A2 arasındaki ilişki

Öteleme

xn1 A1 ile A2 arasındaki X eksenindeki mesafeyi tanımlar.

yn1 A1 ile A2 arasındaki Y eksenindeki mesafeyi tanımlar.

zn1 A1 ile A2 arasındaki Z eksenindeki mesafeyi tanımlar.

Döndürme

A2 bloğunun A1 bloğuna göre A1'in; X,Y,Z eksenlerindeki dönme açılarını tanımlarız.

A1A2A1 kural seti ile çalışacağımız zaman A1 ile A2 arasındaki ilişkiyi ve A2 ile A1 arasındaki ilişkiyi öteleme ve döndürme transformasyonları ile tanımlamamız gerekir.

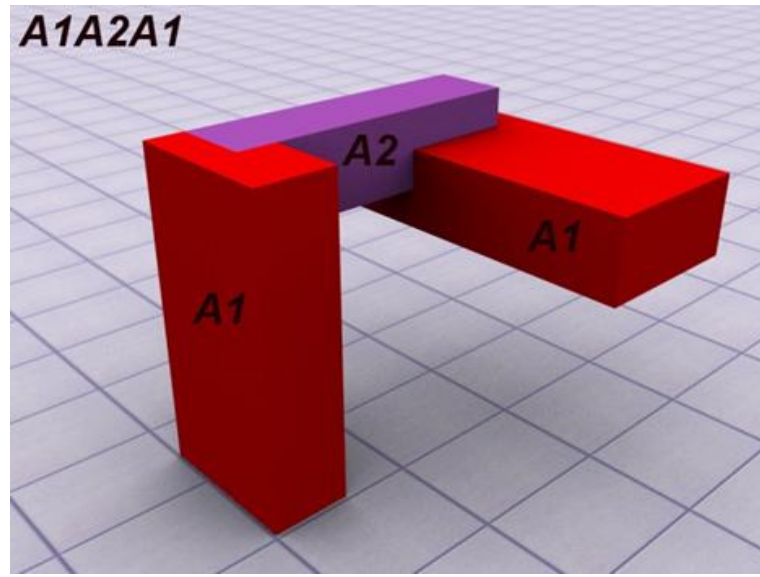
A1A2A3A1 kural seti ile çalışacağımız zaman A1 ile A2 arasındaki ilişkiyi, A2 ile A3 arasındaki ilişkiyi ve A3 ile A1 arasındaki ilişkiyi öteleme ve döndürme transformasyonları ile tanımlamamız gerekir.

A1A2A3A4A1 kural seti ile çalışacağımız zaman A1 ile A2 arasındaki ilişkiyi, A2 ile A3 arasındaki ilişkiyi, A3 ile A4 arasındaki ilişkiyi ve A4 ile A1 arasındaki ilişkiyi öteleme ve döndürme transformasyonları ile tanımlamamız gerekir.

Blok ilişkileri bütün dönüşüm noktalarında yeniden tanımlanabilir, blokların birbirileri ile olan ilişkileri yeniden belirlenebilir.

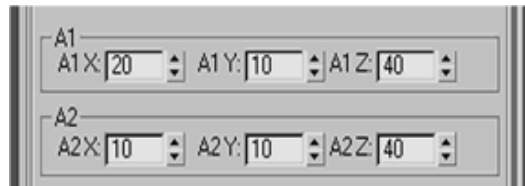
Form Türetici ile Tasarlama

Froebel Form Türetici ile tasarım yaparken tercihe göre kullanılacak blok sayısını kural dizisi seçeneğinden belirledikten sonra blok boyutları belirlenip blok ilişkileri tanımlanıyor. Daha sonra üretmeler yapılıyor. Plug-in, kontrollü olarak, dönüşüm noktalarında tanımlanmış blok parametrelerinin değiştirilmesine imkân vererek, türetilen formları dönüştürmeye (transforme etmeye) olanak sağlıyor. Bu sayede dönüşüm noktalarının belirlenmesi de tasarım aşamasında forma dönüşler yaparak, tasarımın niteliğine göre kullanıcının belirlemesi ile oluşabilecektir.



Şekil 3.4 Tanımlanan blok ilişkileri

Şekil 3.5 'deki blok boyutları, Şekil 3.6 ve Şekil 3.7 deki blok ilişkileri ile Şekil 3.1 de görüldüğü gibi dönüşüm noktası yok seçeneği ve döngü sayısı 1 işaretlenerek oluşturulan A1A2A1 blok dizisi Şekil 3.4'de görülmektedir.



Şekil 3.5 Blok boyutları

Şekil 3.6 A1-A2 arasındaki ilişki

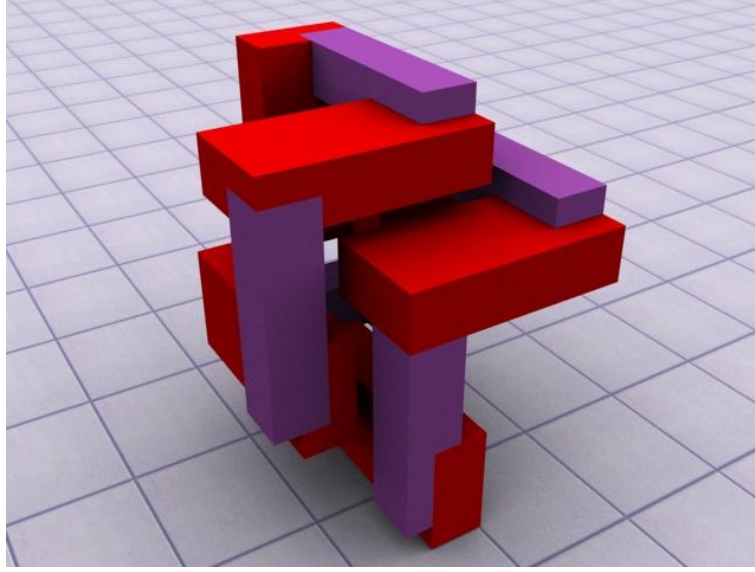
Şekil 3.6'da A2; A1'in Y ekseninde 5 birim ötelenmesi, A1'in Z ekseninde 35 birim ötelenmesi ve A1'in Y ekseninde 90° döndürülmesi ile oluşmaktadır. Böylece A1 ile A2 arasındaki ilişki tanımlanmaktadır.

Şekil 3.7 A2-A1 arasındaki ilişki

Şekil 3.7'de A2'in Y ekseninde 5 birim ötelenmesi, A2'in X ekseninde 5 birim ötelenmesi, A2'in Z ekseninde 35 birim ötelenmesi ve A2'in Y ekseninde 90° döndürülmesi ile A1 oluşmaktadır. Böylece A2 ile A1 arasındaki ilişki tanımlanmaktadır.

Şekil 3.8'deki gibi döngü sayısı 5 yapıлып üret butonuna basıldığında tanımladığımız kural setiyle oluşan form Şekil 3.9'da görülmektedir.

Şekil 3.8 Döngü sayısı 5 ile üretme



Şekil 3.9 Üretilen form aşama 1

Dönüşüm Noktalarının Kullanılması

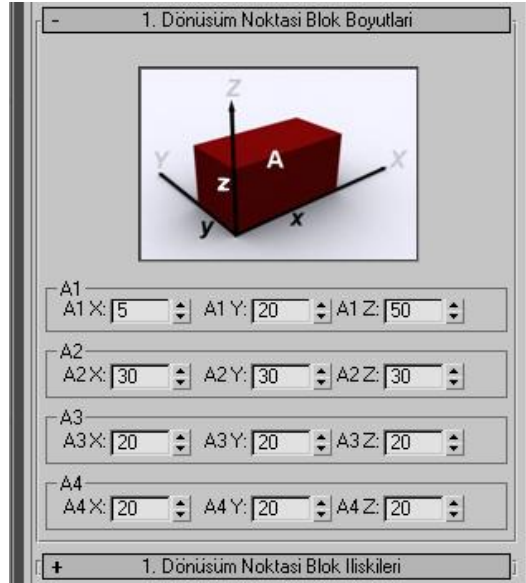
1 Dönüşüm Noktası ile

Öncelikle 1. dönüşüm noktasını kullanmak için 1 dönüşüm noktası seçeneğini seçip döngünün hangi adımda 1. dönüşüm noktası kurallarını alacağını belirlememiz gerekir. Şekil 3.10'da döngü sayımız 10'dur. 1. dönüşüm noktası ise 5'tir. 5 adımdan sonra döngümüz 1 dönüşüm noktası kurallarına göre işleyecektir.



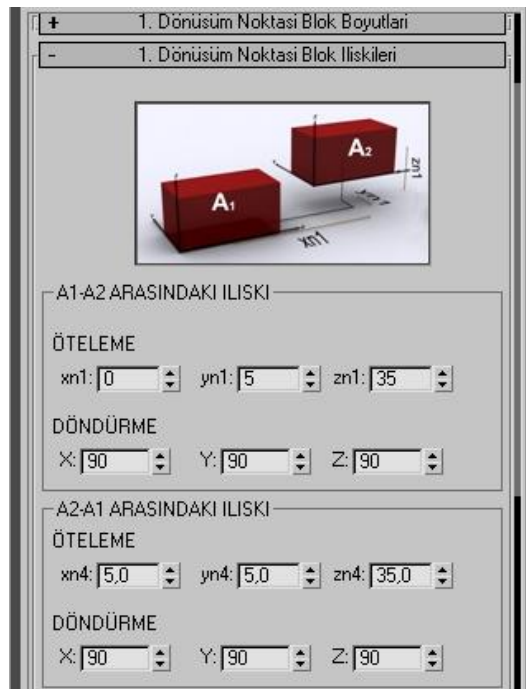
Şekil 3.10 Froebel form üretici 1. dönüşüm noktasını kullanma

Dönüşüm noktası kurallarını ise 1. dönüşüm noktası blok boyutları ve 1. dönüşüm noktası blok ilişkilerini yeniden tanımlayarak belirleyebiliriz.



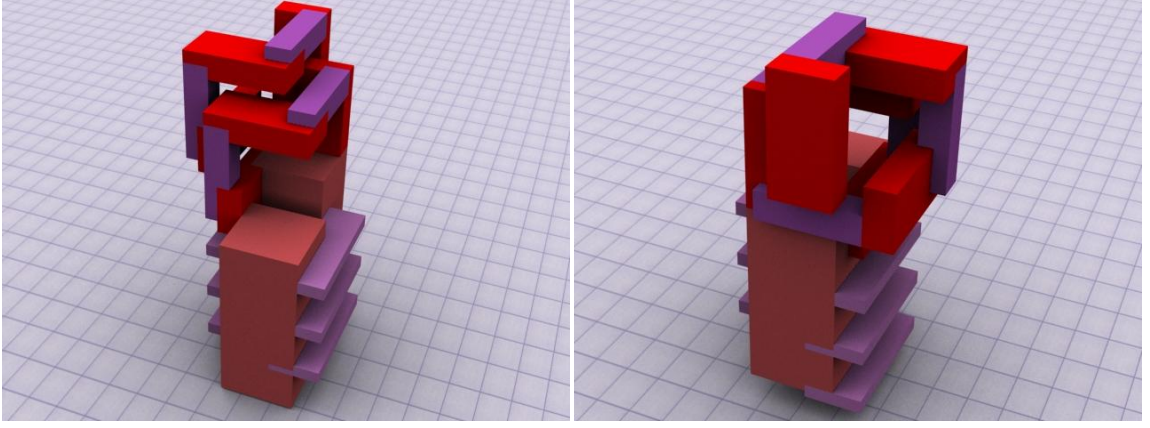
Şekil 3.11 Froebel form üretici 1. dönüşüm noktası blok boyutları

Şekil 3.11 ve Şekil 3.12'deki gibi 1. dönüşüm noktası blok boyutları ve 1. dönüşüm noktası blok ilişkileri değiştirilebilir. Blok ilişkileri ve blok boyutları yeniden tanımlanırken üret butonuna basarak denemeler yaparak en uygun blok boyutları ve blok ilişkileri belirlenir. Bu kararlara sezgiler ve öngörüler yön verecektir.



Şekil 3.12 Froebel form üretici 1. dönüşüm noktası blok ilişkileri

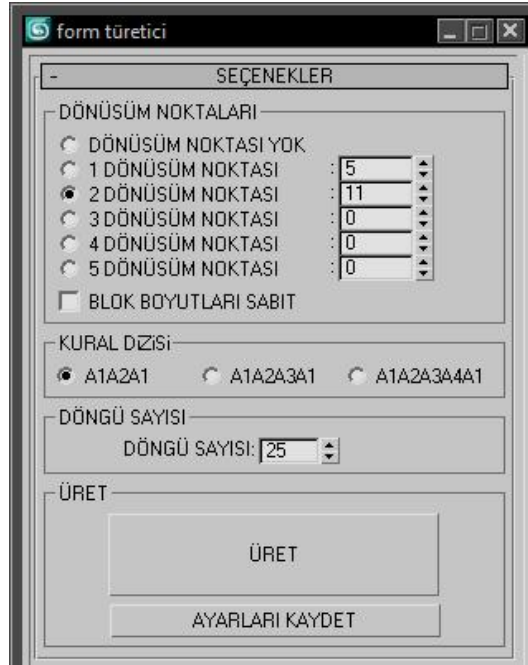
Bir dönüşüm noktası ile denemeler yapıp şekil 3.11 ve şekil 3.12'deki değerlere ulaşarak türettiğimiz formlara ait perspektifler Şekil 3.13'teki gibidir.



Şekil 3.13 Üretilen form aşama 2

Perspektiflerde görüldüğü gibi 5 adım boyunca başlangıç blokları ve blok ilişkileri ile süren döngü 5. adımdan sonra yeni tanımladığımız blok boyutlarına ve blok ilişkilerine göre devam etmektedir.

2 Dönüşüm Noktası ile

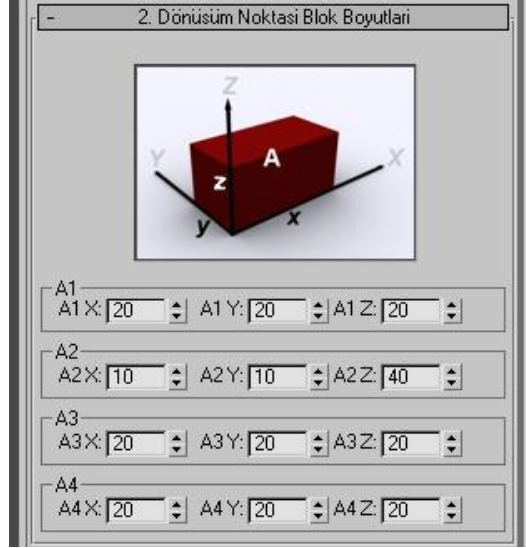


Şekil 3.14 Froebel form üretici 2. dönüşüm noktası kullanma

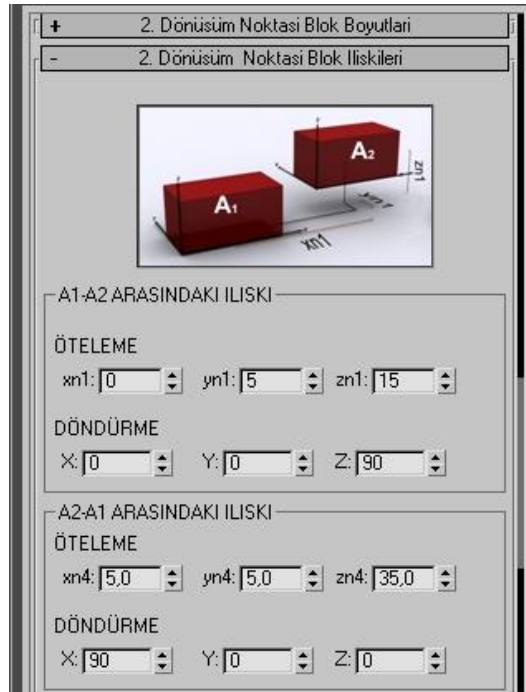
Öncelikle 2. dönüşüm noktasını kullanmak için 2 dönüşüm noktası seçeneğini seçip döngünün hangi adımda 1. dönüşüm noktası kurallarını ve 2. dönüşüm noktası kurallarını alacağını belirlememiz gerekir. Şekil 3.14'de döngü sayımız 25'dir. 1. dönüşüm noktası 5, 2. dönüşüm

noktası 11'dir. 5 adımdan sonra döngümüz 1 dönüşüm noktası kurallarına göre işleyecektir. 11. adımdan sonra ise döngümüz 2. dönüşüm noktası kurallarına göre işleyecektir.

2. dönüşüm Noktası Kurallarını ise 2. dönüşüm noktası blok boyutları ve 2. dönüşüm noktası blok ilişkilerini yeniden tanımlayarak belirleyebiliriz.

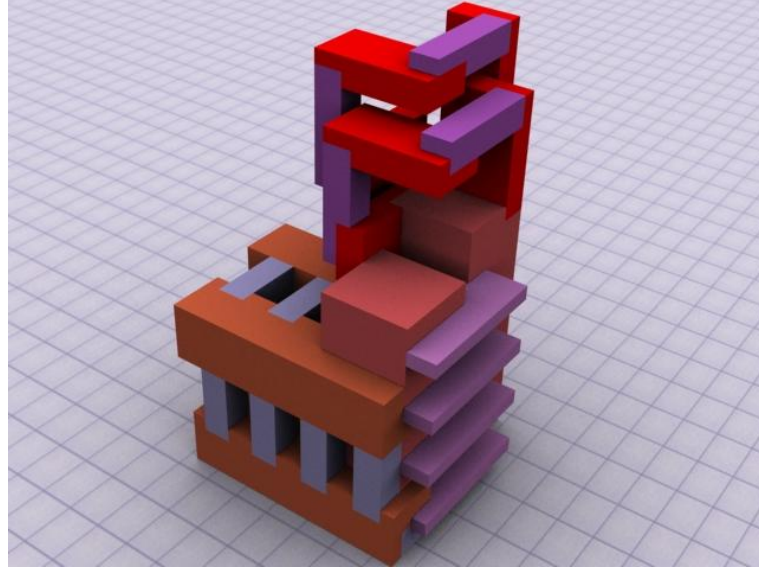


Şekil 3.15 Froebel form üretici 2. dönüşüm noktası blok boyutları

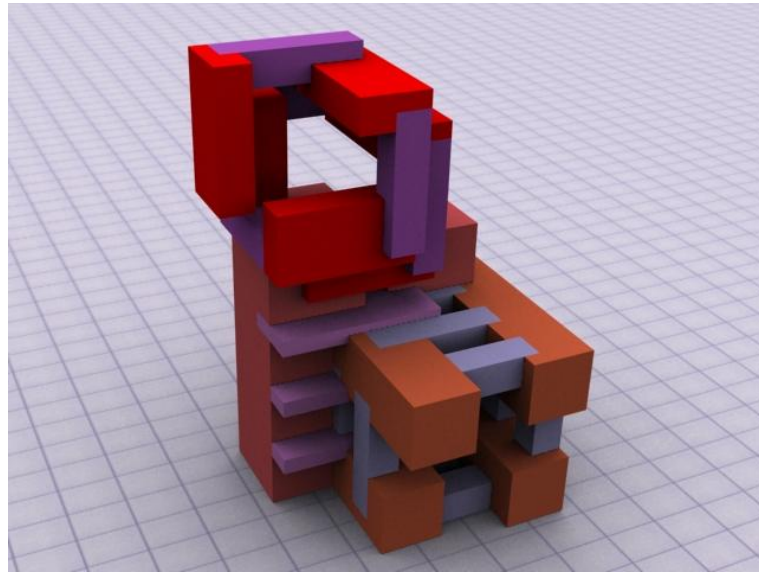


Şekil 3.16 Froebel form üretici 2. Dönüşüm noktası blok ilişkileri

1 dönüşüm noktası seçeneğinde yaptığımız gibi 2 dönüşüm noktası seçeneğinde de blok boyutlarını belirledikten sonra bloklar arasındaki ilişkileri belirlerken türetmeler yapıp formun gelişimini gözleyip uygun ilişkinin seçimini yapabiliriz.



Şekil 3.17 Üretilen form aşama 3



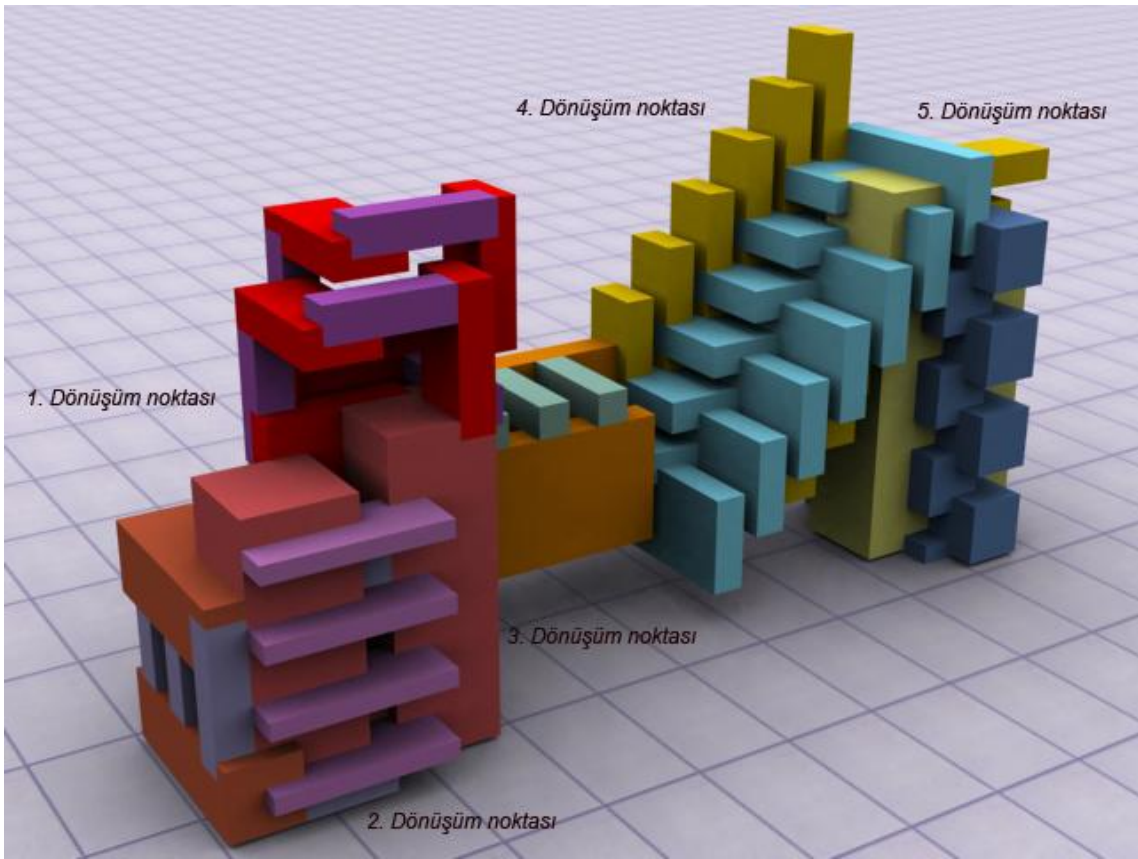
Şekil 3.18 Üretilen form aşama 3

Şekil 3.17 ve Şekil 3.18’de görüldüğü gibi 5 adım boyunca başlangıç blokları ve blok ilişkileri ile süren döngü 5. adımdan sonra 11. adıma kadar 1. dönüşüm noktası blok boyutları ve blok ilişkileri ile sürmektedir. 11. adımdan sonra ise 2. dönüşüm noktası blok boyutları ve blok ilişkileri olarak yeniden tanımladığımız değerlere göre döngümüz devam etmektedir.

5 Dönüşüm Noktası ile

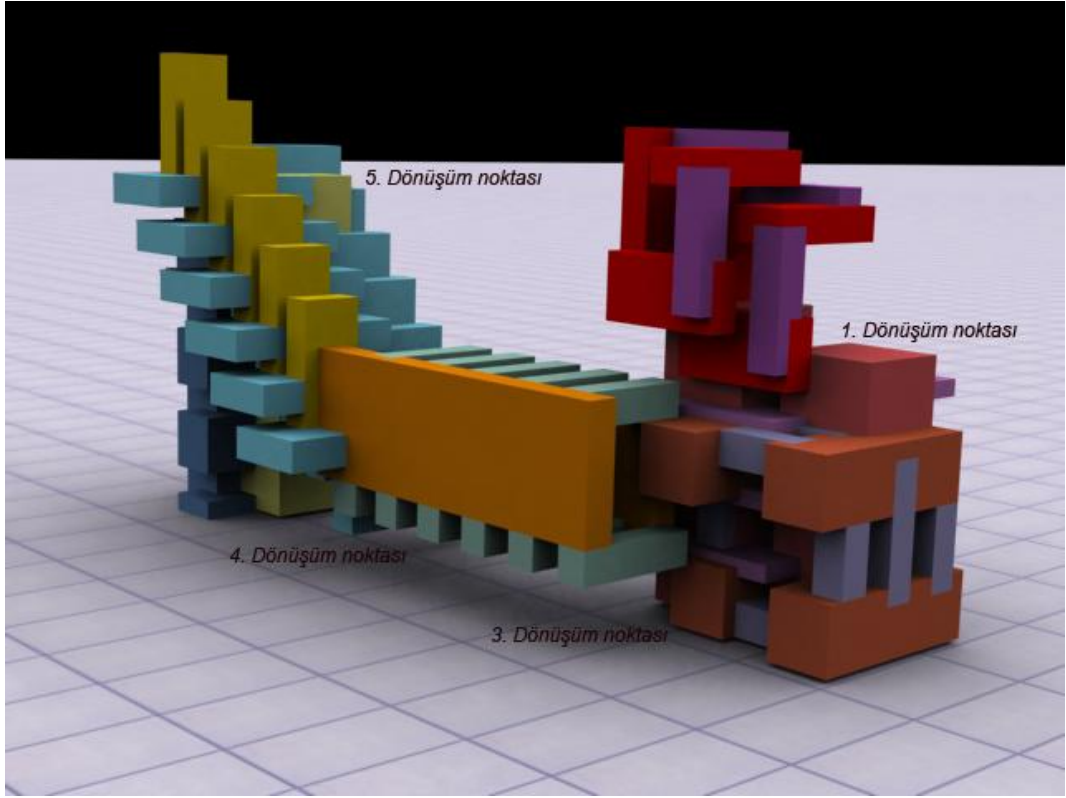
Şekil 3.19 ve Şekil 3.20’de 5 dönüşüm noktası seçeneği kullanılarak türetilmiş form görülmektedir. Her dönüşüm noktasından sonra form 2 farklı renk alarak blok boyutlarındaki ve blok ilişkilerindeki değişikliklerin gözlenmesi amaçlanmıştır.

1. ve 2. dönüşüm noktası için belirlediğimiz blok boyutları ve blok ilişkileri gibi 3-4 ve 5. adımlar için de yeni blok boyutları belirleyip ilişkilerini tanımlarız. İlişkileri belirlerken üret butonuna basıp oluşan formu görüp istediğimiz dönüşüm noktasında blok boyutları ve blok ilişkilerine müdahale edip formun oluşumunu yönlendirebiliriz.



Şekil 3.19 5 dönüşüm noktası kullanılarak üretilen form

Şekil 3.19’da görüldüğü üzere bütün dönüşüm noktalarında bloklar, blokların birbirileri ile olan ilişkileri değişebilmektedir. Her dönüşüm noktasının adım sayısı blok özellikleri ve blok ilişkileri geri dönülüp değiştirilip oynamalar ile form arayışı yapılabilir. Burada amaç tasarım sürecinde tasarımcının yorumlayabileceği bir formu sezgisel olarak geliştirip yönlendirmesidir. Süreç rastgele denemeler yaparak form arama şeklinde de devam edebilir.



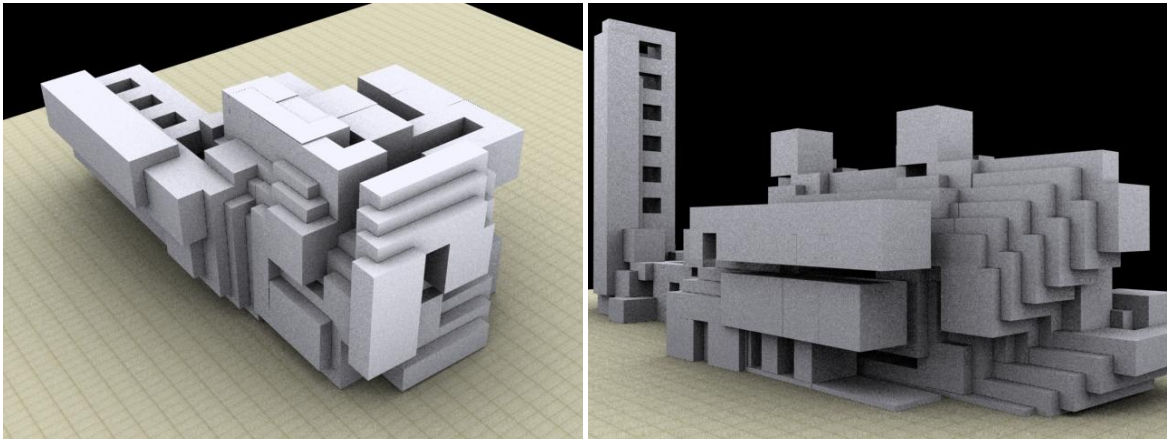
Şekil 3.20 5 dönüşüm noktası kullanılarak üretilen form

Örnek Form bulma çalışmaları

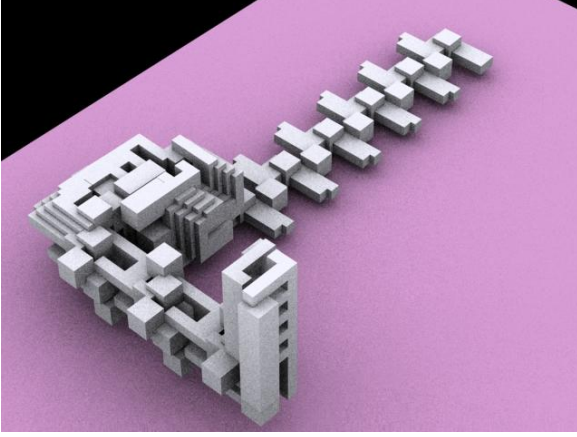
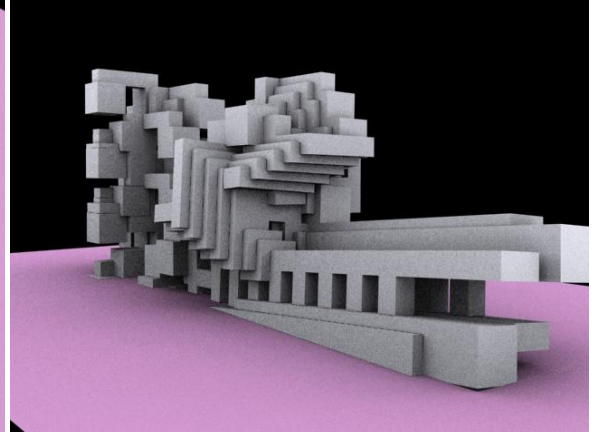
Aşağıda A1A2A3A4A1 kural dizisi seçeneği ve 5 dönüşüm noktası kullanarak elde edilmiş formlar bulunmaktadır. Formlar elde edilirken dönüşüm noktalarındaki blok boyutları ve blok ilişkileri değerleri değiştirilerek denemeler yapılmıştır.

Deneme 1

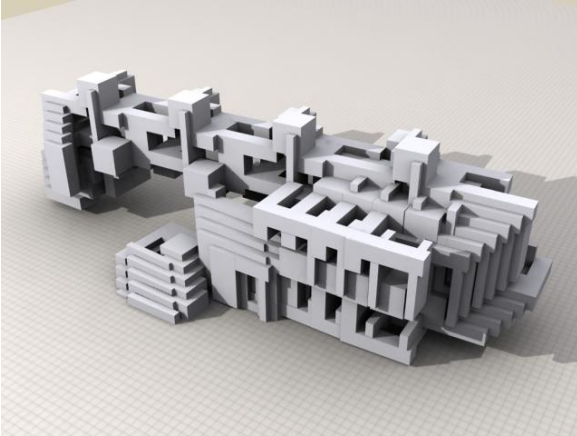
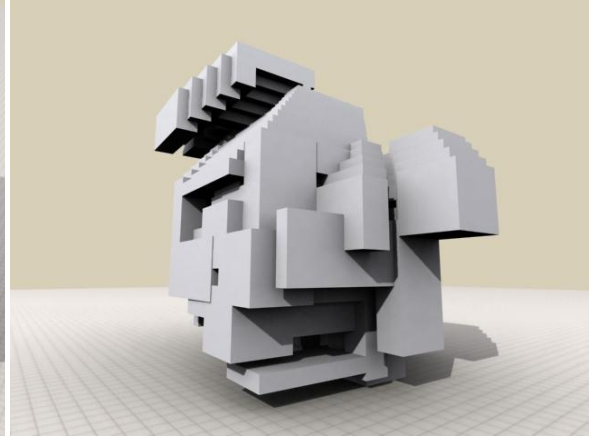
Deneme 2



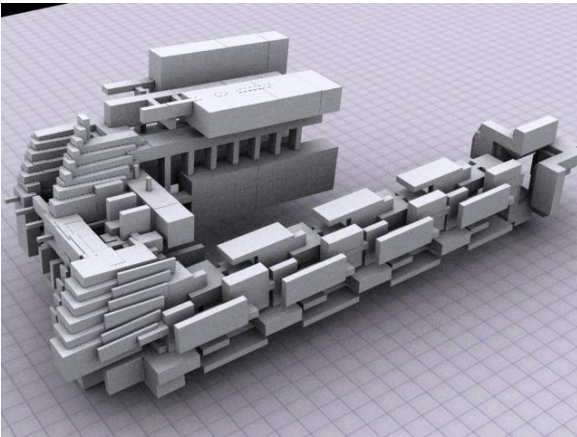
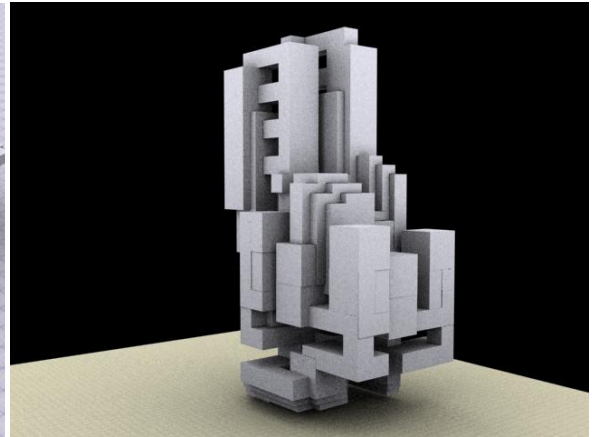
Şekil 3.21 Form türetme denemesi-1,2

Deneme 3*Deneme 4*

Şekil 3.22 Form türetme denemesi-3,4

Deneme 5*Deneme 6*

Şekil 3.23 Form türetme denemesi-5,6

Deneme 7*Deneme 8*

Şekil 3.24 Form türetme denemesi-7,8

3.4. Sentez

Bütün ortamlarda çalışmamaktadır. Türeticinin çalışması için 3dsMax programı gereklidir.

A1, A2, A3, A4 olarak belirlenen, aralarındaki ilişkilerin kullanıcı tarafından tanımlandığı ve kural setlerinin oluşturulduğu blokların sayısı daha da artırılabilir.

Tasarım uzayını daha da artırmak için dönüşüm noktaları sayısı artırılabilir.

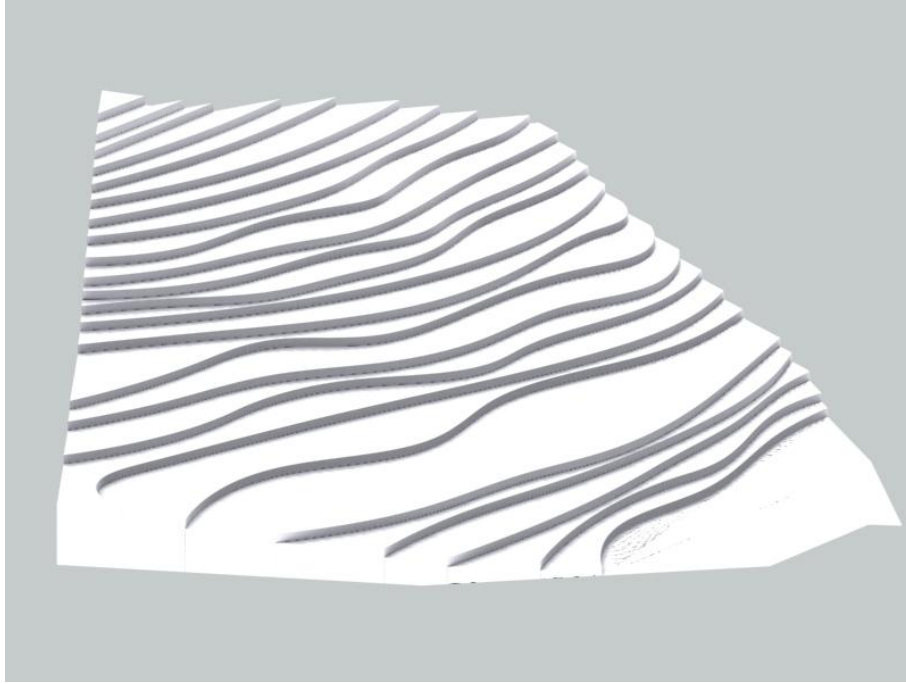
Blok parametreleri ve bloklar arası ilişkiler, çek-bırak (drag and drop) yöntemiyle belirlenememesi ve değer girilerek oluşturulması kullanım kolaylığını kısıtlamaktadır. Bu sorunun halledilebilmesi ile arayüzün sadeleşmesi, blok sayısı ve dönüşüm noktası gibi seçeneklerin artırılmasına imkan verebilecektir.

Türetici ile denemeler yapılırken, ekranın temizlenip yeni denemelere hazır hale getirilmesi, gerektiğinde kullanıcı tarafından seçilip silinmesi ile gerçekleştirilmektedir. Bu doğrultuda ekranı temizle butonu geliştirilebilir.

Son çalışılan ayarlar ile daha sonra çalışmaya devam edilebilmesi için mevcut ayarları kaydet seçeneği aktif hale getirilebilir. Mevcut durumunda bunun gerçekleşmesi için kullanıcının son ayarlarının resmini kaydedip, daha sonra tekrar çalışmaya başladığında aynı parametreleri tekrar girmesi gerekmektedir.

Blok boyutlarının bütün dönüşüm noktalarında sabit kalması istediğinde, blok boyut parametrelerini bütün dönüşüm noktalarına aktaracak buton aktif hale getirilebilir.

4. GRAMER TABANLI KONUT YERLEŐME TASARIMI



Őekil 4.1 Konut yerleŐmesinin tasarlanacađı arsa

4.1 Tasarım S¼recini Tasarlama

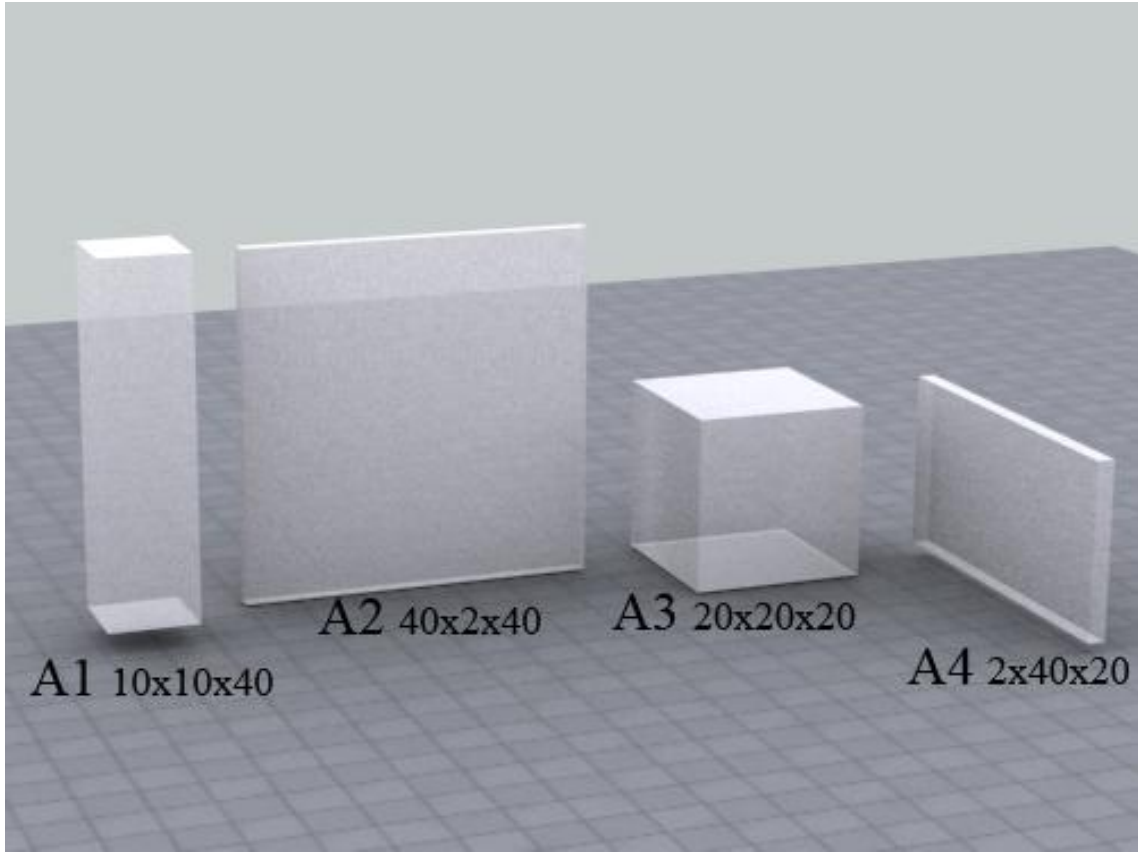
Konut yerleŐmesi tasarımının uygulanacađı arsa hipotetik olarak belirlenmiŐtir. Arsada tasarlanması ¼ng¼r¼len konut tiplerinin ¼ç ayrı tipten oluŐmasına ve toplamda 10 adet olmasına karar verilmiŐtir. Arazi yapısı ve diđer etmenler d¼Ő¼n¼lerek form t¼retmeleri y¼nlendirilmiŐtir. Buna g¼re konut tipleri eđimin artışı ve y¼neliŐini g¼re yorumlanacaktır. T¼retmeler yapılırken oluŐabilecek formların olası alanlarının imar durumu kıstaslarına uygun ¼lçeklerde olmasına ¼zen g¼sterilmiŐtir. Bu dođrultuda kullanılacak formların boyutları sečilirken, oluŐturacakları mek¼nlarında ¼ng¼r¼len ¼lçeklerde olması amaçlanmıŐtır.

4.2 Scripting

Konut yerleŐmesinin tasarım s¼recinde kullanılan script, daha ¼nceki baŐlıkta ele alınan Froebel form t¼reticidir. Froebel form t¼retici ile konut tipleri i¼in kullanılacak formun belirlenmesi amaçlanmıŐtır. Froebel form t¼retici ile mevcut verilere g¼re seçilen blok boyutları girilip, blokların birbirleri ile olan iliŐkileri deđerlendirilip arazi ve diđer koŐullar d¼Ő¼n¼lerek t¼retmelerin yapılması hedeflenmiŐtir.

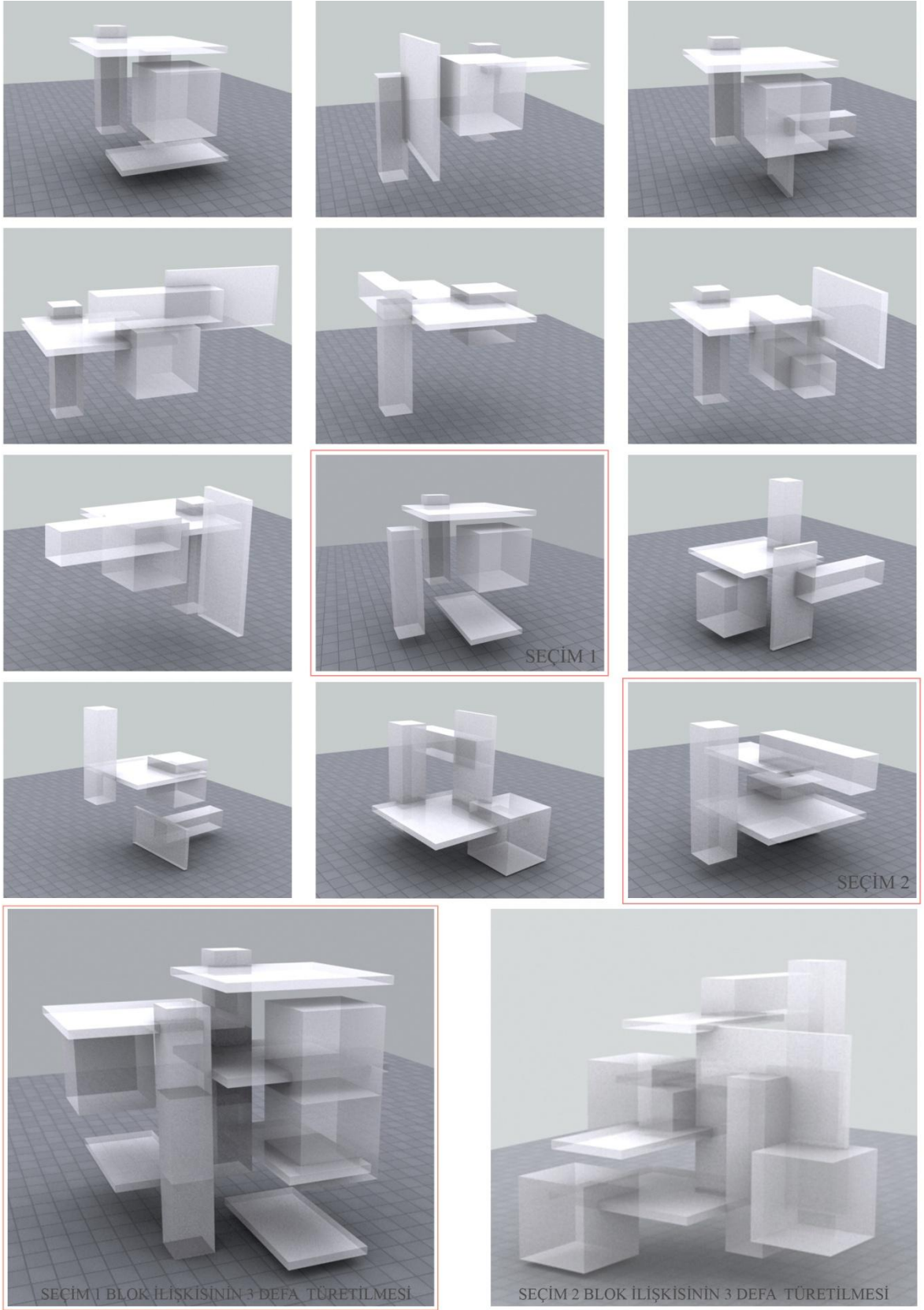
4.3 Kural Setleri ve Kural Tanımlama

Froebel form üretici ile konut tasarımı için 3 boyutlu eskizler elde etmek üzere üretme çalışmalarına başlarken öncelikle froebel form üreticide tanımladığımız gibi kaç blok kullanacağımızı seçeriz. Kullanacağımız blokların boyutlarını froebel form üreticiye girerek blokların birbirileri ile olan ilişkilerini üretmeler ve denemeler yaparak değerlendirip blok boyutları ile ilgili seçimimizi sezgisel olarak, anlam yükleyerek ya da rastgele yaparız. Aşağıda konut tasarımı için seçilen 4 adet blok bulunmaktadır.



Şekil 4.2 Seçilen bloklar

Yukarıda seçilmiş olan bloklar, form üreticinin blok boyutları bölümüne yazılmıştır. Blokların birbirileri ile olan ilişkilerini düzenlemek için froebel form üreticinin blok ilişkileri bölümüne blokların birbirileri ile olan ilişkileri girilmiştir. Çeşitli değerler vererek ve denemeler yapılarak A1A2A3A4A1 blok kural dizisiyle oluşan formlardan bazıları aşağıdaki şekilde listelenmektedir. Oluşan formlardan birini seçerek başlangıç formun kurallarının belirlenmesi kullanıcının öngörülerini doğrultusunda olmaktadır. Bu seçim yapılırken kural setinin döngü sayısı artırıldığında oluşacak formlarda gözlenebilmektedir. Aşağıdaki şekilde elde edilen formlardan 12 tanesi ele alınmıştır.

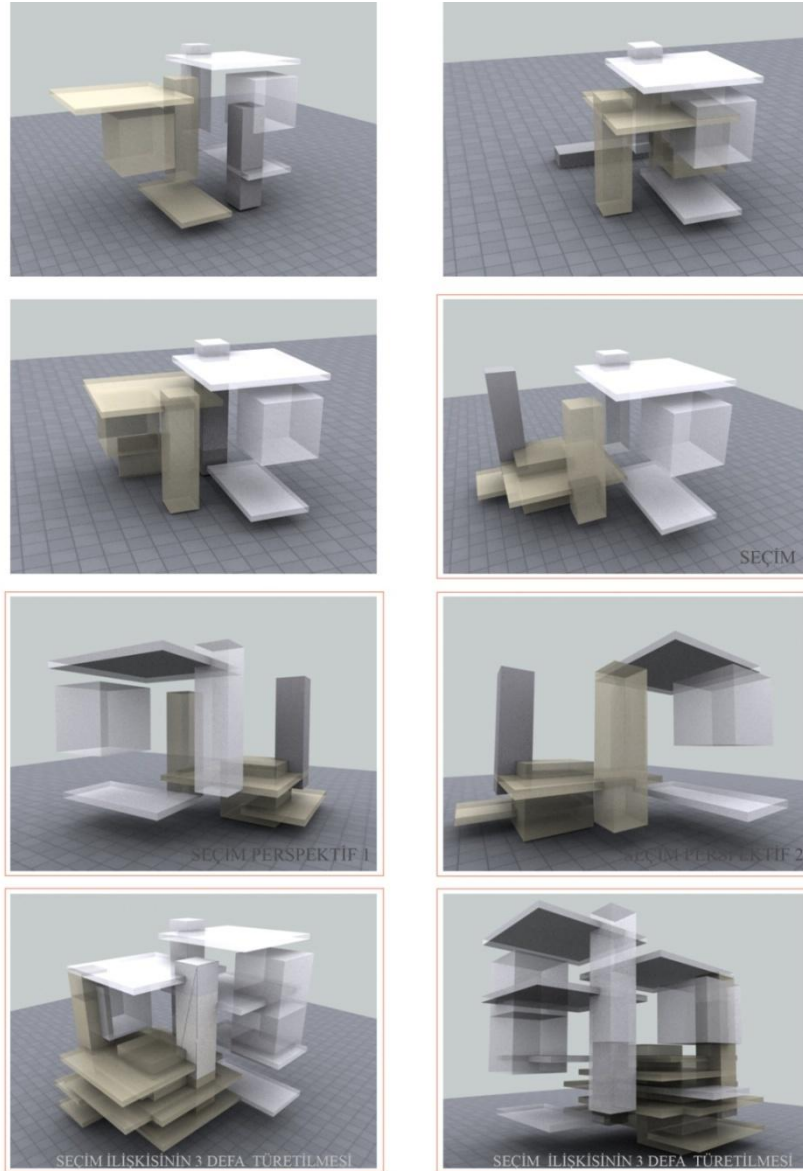


Şekil 4.3 Bloklar arası ilişki seçenekleri

4.4 Tasarımcının Müdahalesi

Yukarıdaki şekilde üretilen formlardan seçim 1 ve seçim 2 ele alınmıştır. Seçimler döngü sayısı 3 yapılarak türetildiğinde elde edilen formlar gözlenerek seçim 1'e karar verilmiştir.

Seçilen form, froebel form üreticinin başlangıç blok boyutları ve blok ilişkileridir. Türetilmiş olan form kullanıcı tarafından tanımlanan 1. dönüşüm noktasındaki bloklar arası ilişkilerin yeniden tanımlanmasıyla geliştirilecektir. Şekil 4.4'te ilk 4 resimde 4 farklı ilişki tanımlanmış ve döngü sayısı artırılıp denemeler yapılmıştır.

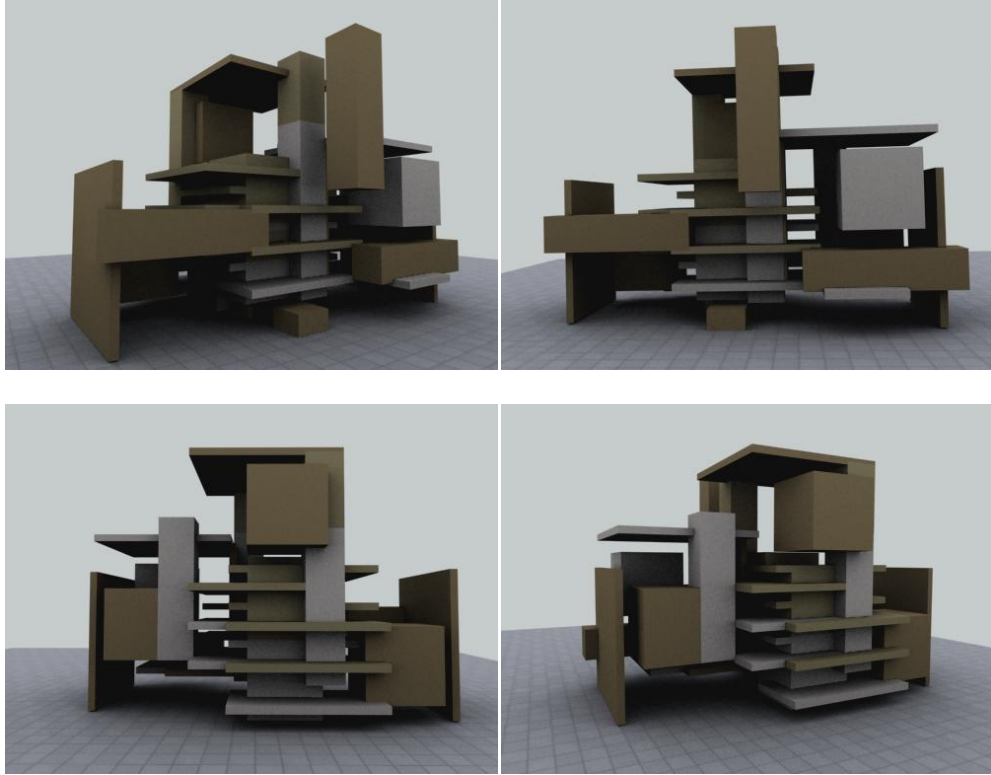


Şekil 4.4 1 Dönüşüm noktası kullanılarak türetilen seçenekler ve seçilen form

4.5 Form Oluşturma

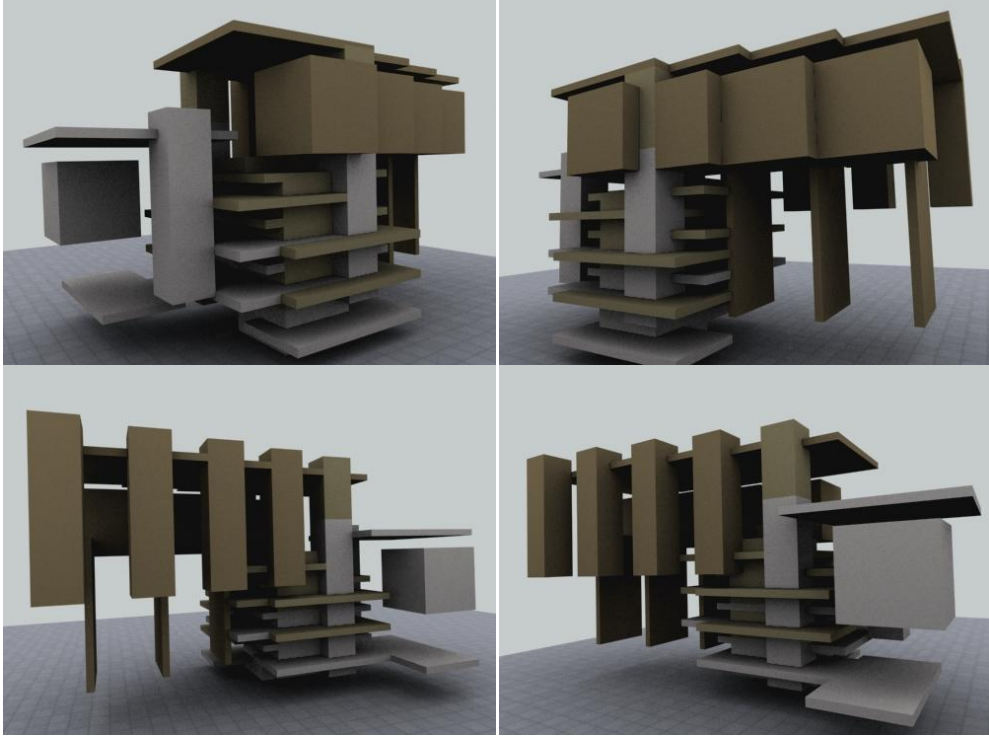
Yukarıdaki şekilde başlangıç blok ilişkileri beyaz, 1. dönüşüm noktası blok ilişkileri ise bej renktedir. Bej grubu ile beyaz grup arasından en iyi uyum gösteren türetme döngü sayısı artırılıp denemeler yapılarak seçilmiştir. Yapılan türetmelerden sonra ele alınan form kompozisyonunun konut tasarımı sürecinde ele alınmasına karar verilmiştir. Froebel form türetici ile başlangıç ve 1. dönüşüm noktası ilişkileri belirlenmiştir. Form arayışına 2. dönüşüm noktası kurallarının yeniden tanımlanmasıyla devam edilmiştir. 2 dönüşüm noktası kullanarak konut tasarımı için form eskizlerini elde edinilebileceği öngörülmüştür. Bu durumda başlangıç blok ilişkileri ve 1. dönüşüm noktası blok ilişkileri sabit kalarak, 2 dönüşüm noktası blok ilişkilerinde değişiklikler yapılmış ve türetilmiştir. Türetmelerle oluşan formlar arasından Şekil 4.5, 4.6, 4.7, 4.8, 4.9, 4.10, 4.11, 4.12, 4.13 ve 4.14'te gösterilen 10 seçenek ele alınmıştır.

Seçenek 1



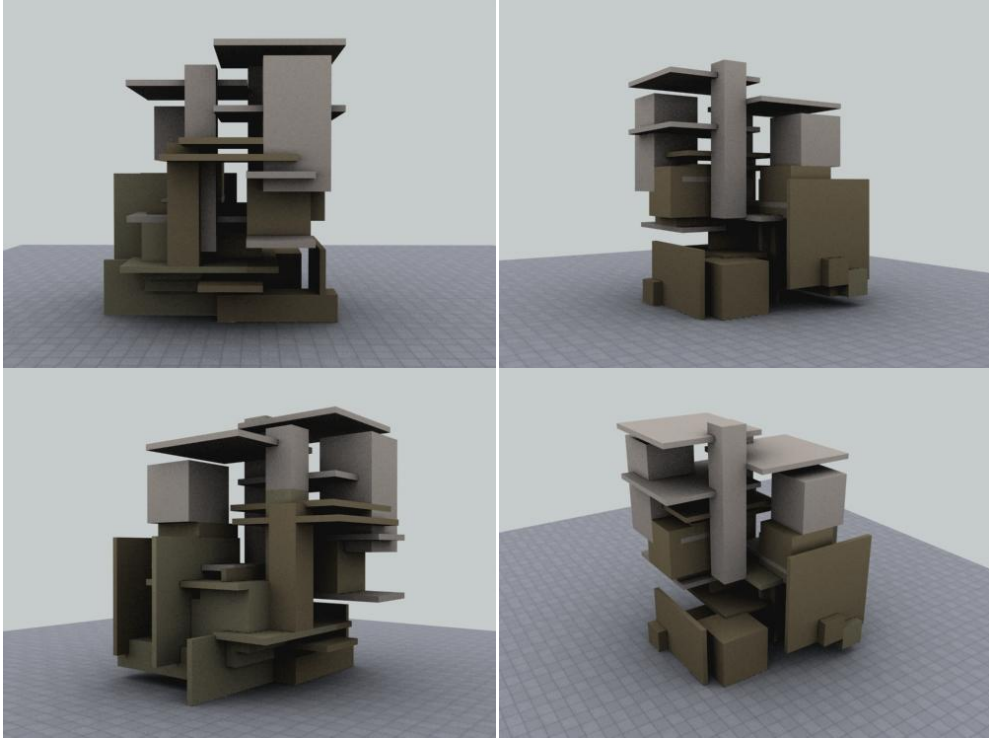
Şekil 4.5 2 Dönüşüm noktası kullanılarak türetilen form-1

Seenek 2

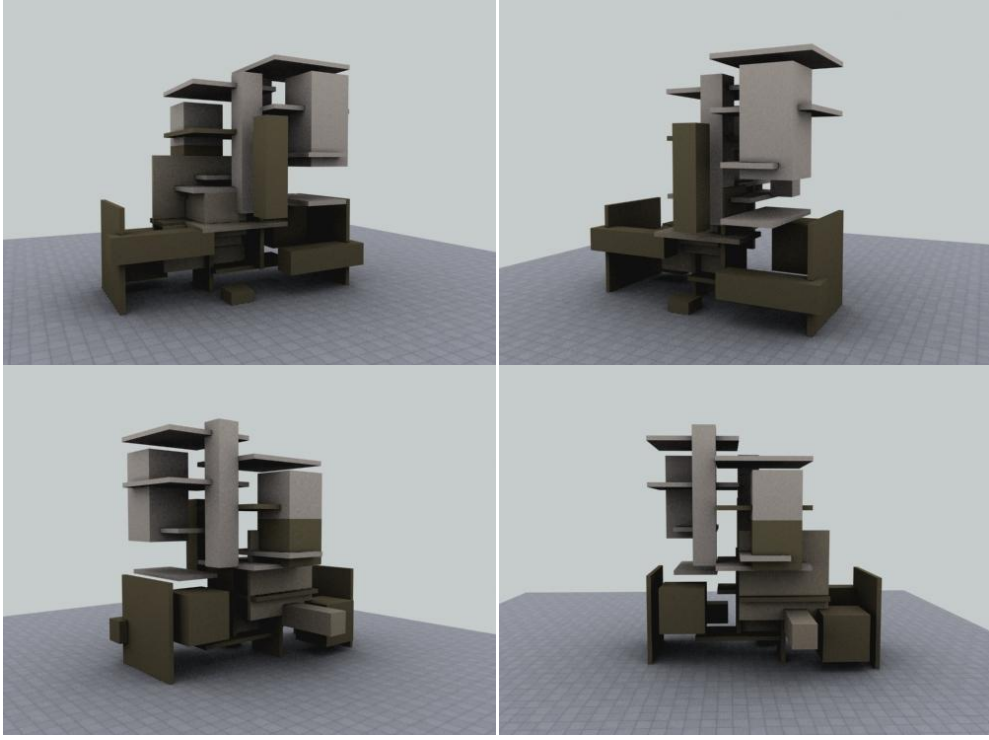


Şekil 4.6 2 Dönüşüm noktası kullanılarak türetilen form-2

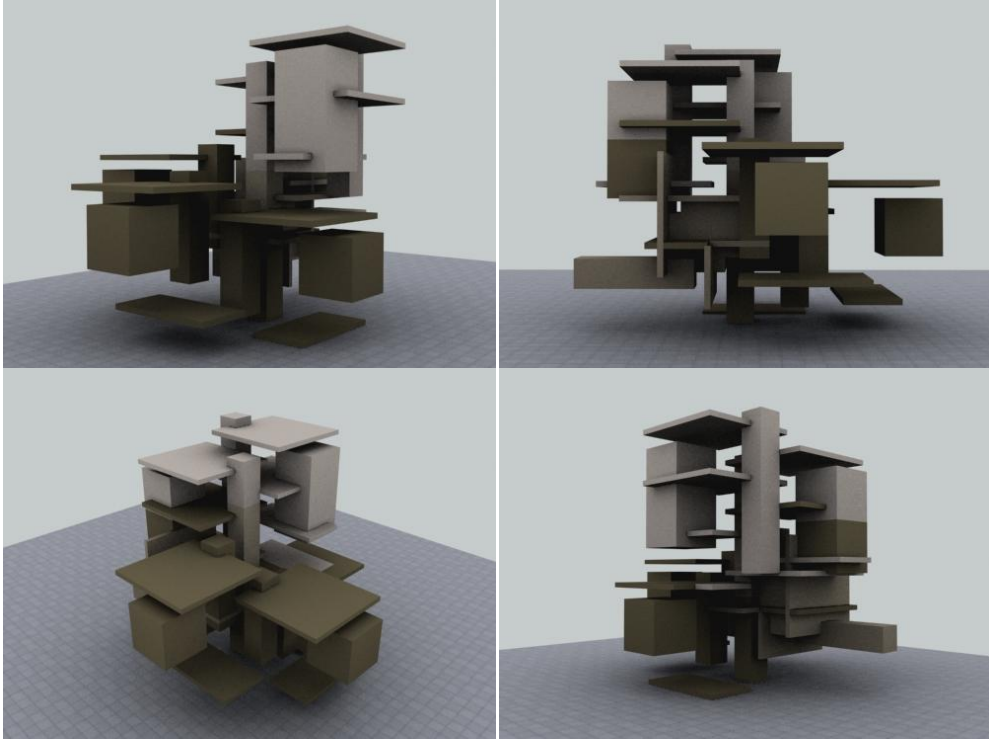
Seenek 3



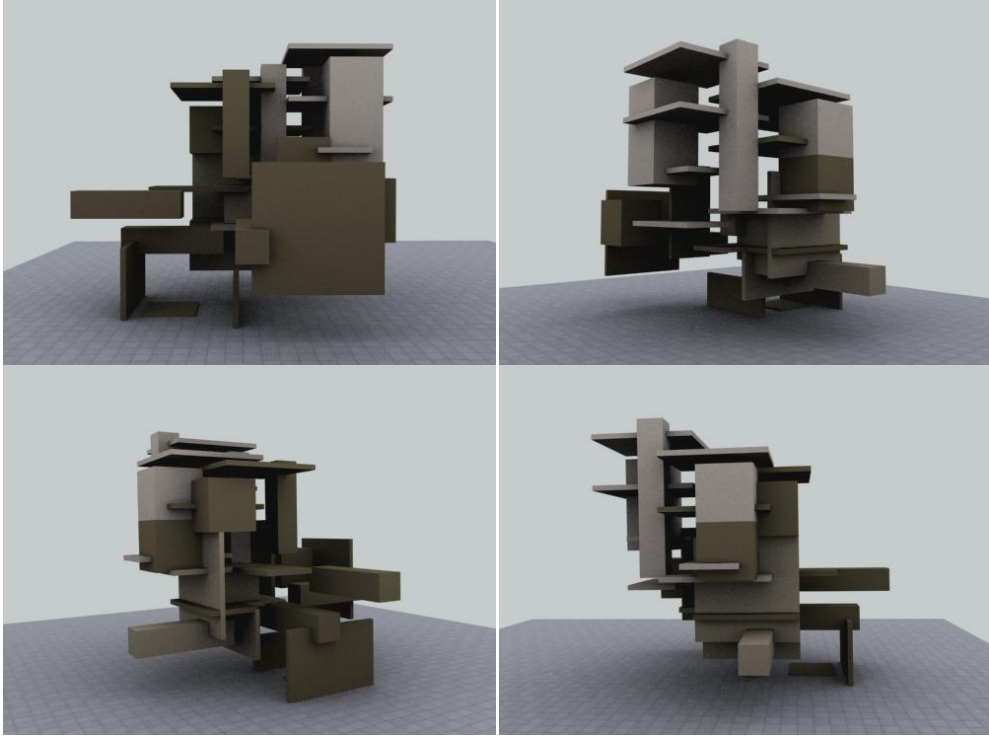
Şekil 4.7 2 Dönüşüm noktası kullanılarak türetilen form-3

Seenek 4

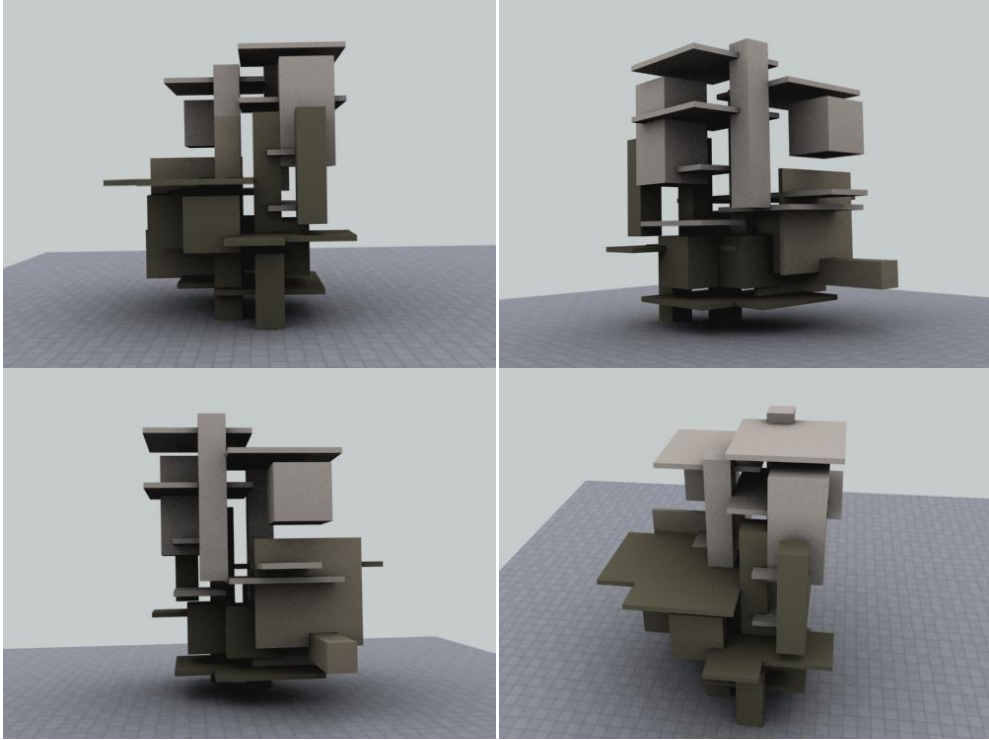
Şekil 4.8 2 Dönüşüm noktası kullanılarak türetilen form-4

Seenek 5

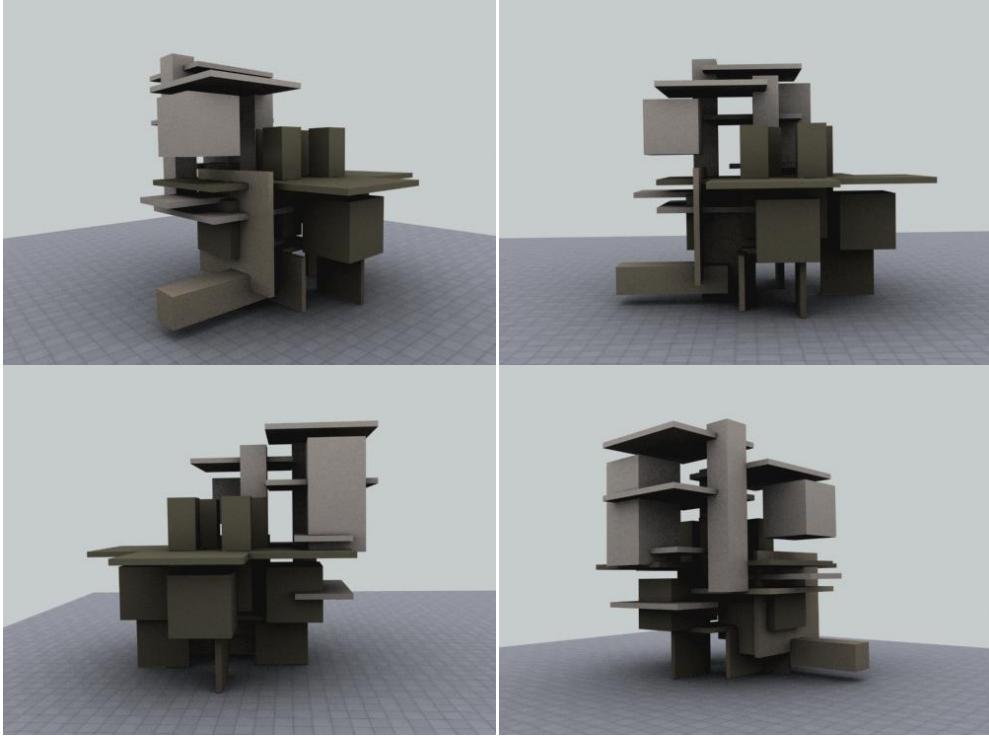
Şekil 4.9 2 Dönüşüm noktası kullanılarak türetilen form-5

Seenek 6

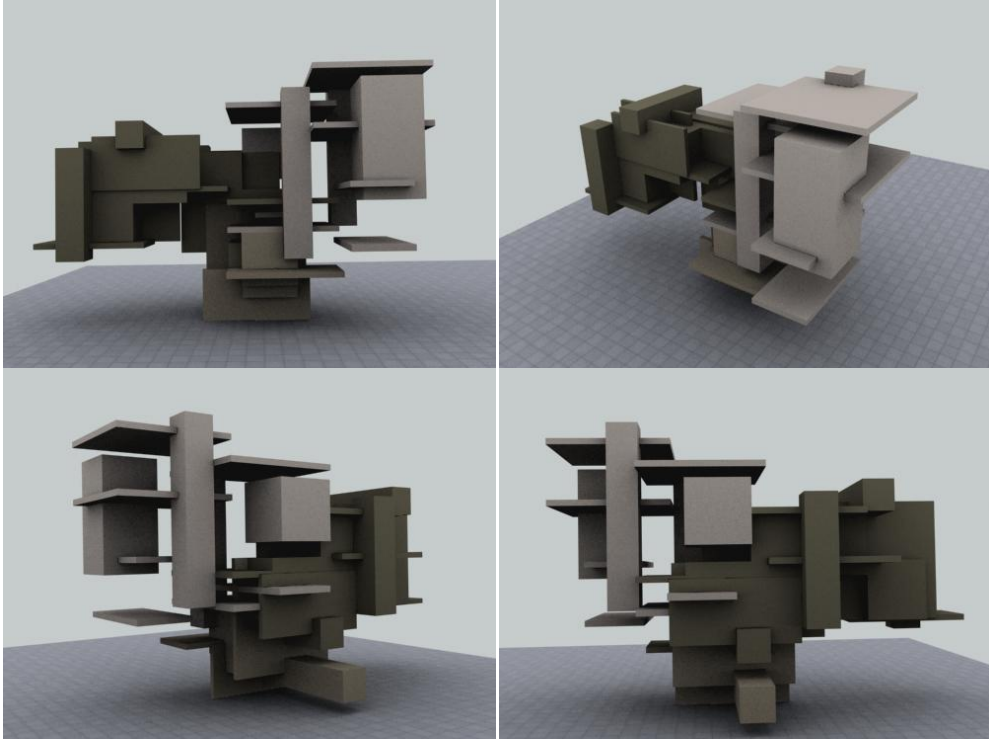
Şekil 4.10 2 Dönüşüm noktası kullanılarak türetilen form-6

Seenek 7

Şekil 4.11 2 Dönüşüm noktası kullanılarak türetilen form-7

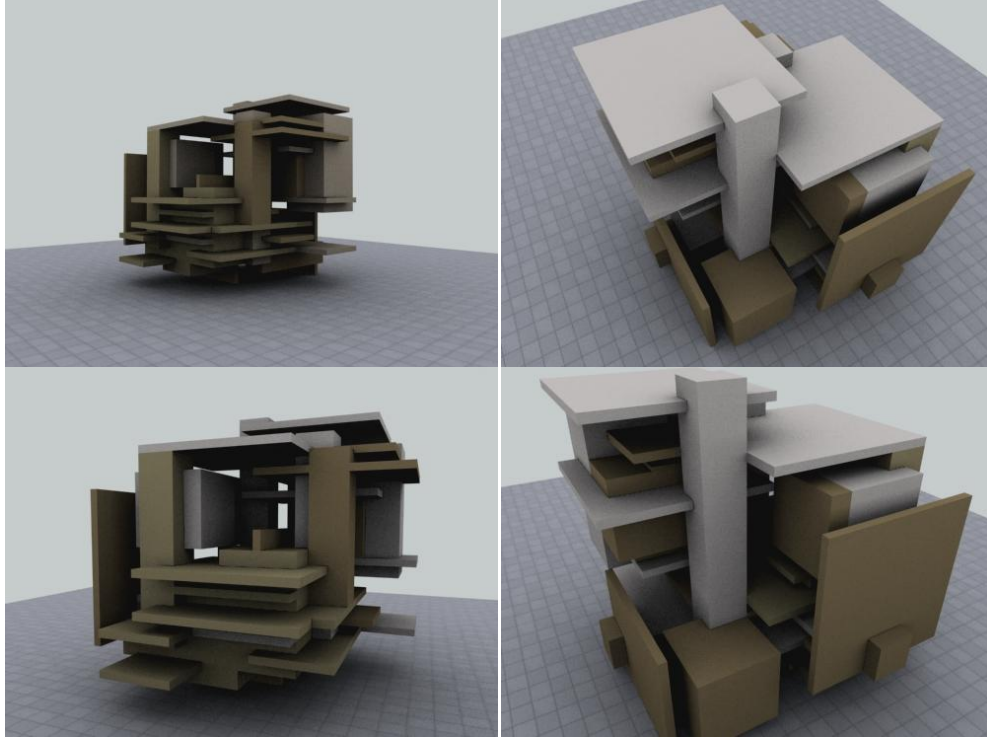
Seenek 8

Şekil 4.12 2 Dönüşüm noktası kullanılarak türetilen form-8

Seenek 9

Şekil 4.13 2 Dönüşüm noktası kullanılarak türetilen form-9

Seçenek 10

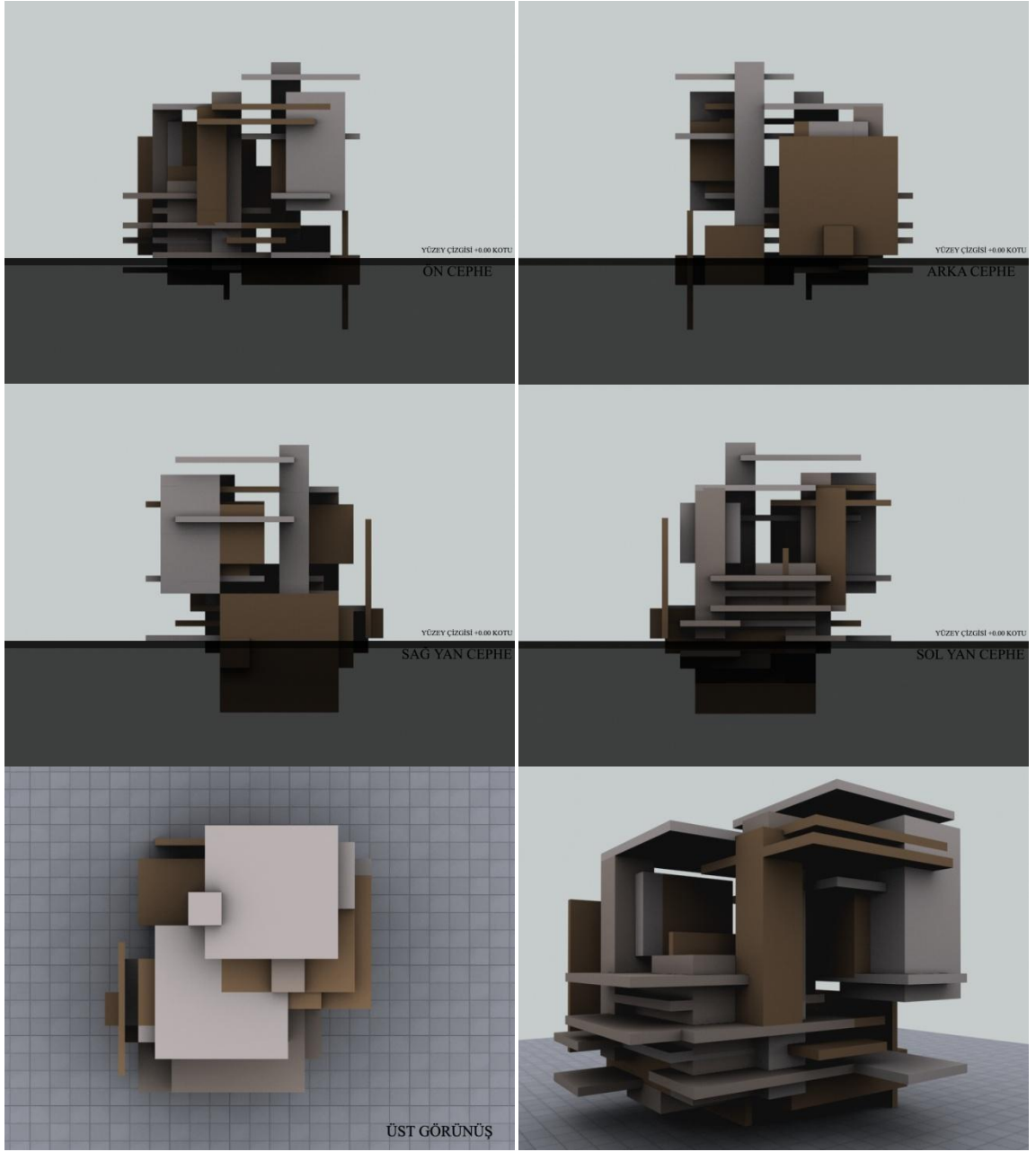


Şekil 4.14 2 Dönüşüm noktası kullanılarak türetilen form-10

4.6 Formun Yorumlanması

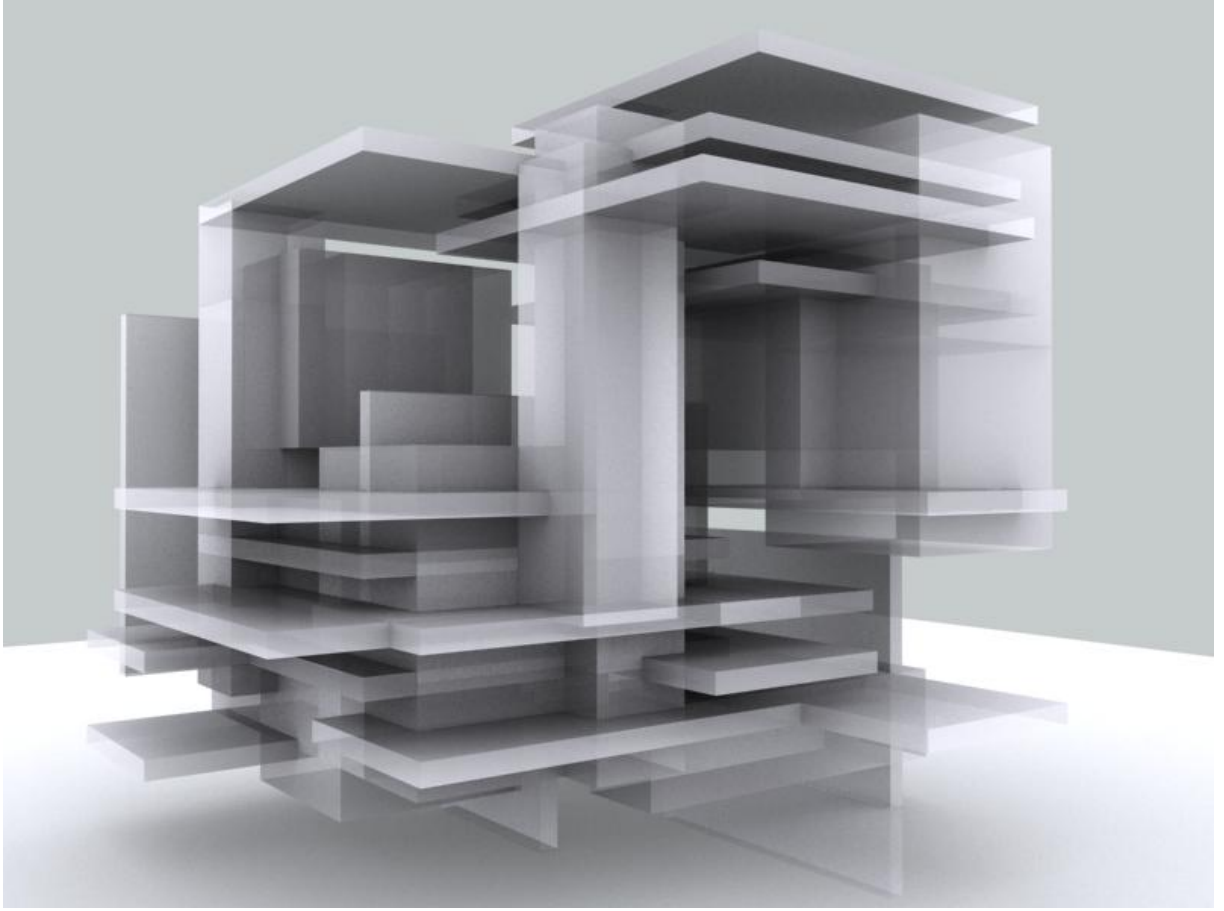
Bu çalışma gramer tabanlı tasarımın konut projelerinde uygulanabilirliğini denemek amacıyla yapılmıştır. Tasarım sürecine başlarken tasarımcının öngördüğü kıstaslar sadece arazi verileridir. Gerçek bir konut uygulama projesinde ise bu kıstaslar çevre, arazi, malzeme, maliyet, yapı, kullanıcı talepleri gibi daha geniş bir listeyi içerecektir.

Görünüş ve perspektifleri incelenerek değerlendirilen formlar arasından, tasarım sürecine başlarken verilen kıstaslara en uygun görülen 3 adet form üzerinde çalışılmaya karar verilmiştir. Gerçek bir uygulamada tasarım süreci yukarıda belirttiğimiz kıstaslara göre ele alınacaktır. Seçenek 1, Seçenek 5 ve Seçenek 10 ile oluşan formlar uygun görülüp görünüşleri, kesitleri, plan kesitleri incelenerek formun somutlaştırılması, bir konut projesine dönüşmesi için gerekli müdahaleler yapılmıştır. Formun yorumlanması tasarımcıdan tasarımcıya fark edebilmektedir. Tasarımcı, oluşan formları 3 boyutlu eskizler olarak değerlendirip, formu kendi tasarım kıstasları ile geliştirmelidir. Aşağıda sırasıyla seçenek 10, seçenek 1 ve Seçenek 5 türetmelerinin değerlendirme çalışmalarının süreçleri bulunmaktadır.

SEÇENEK 10

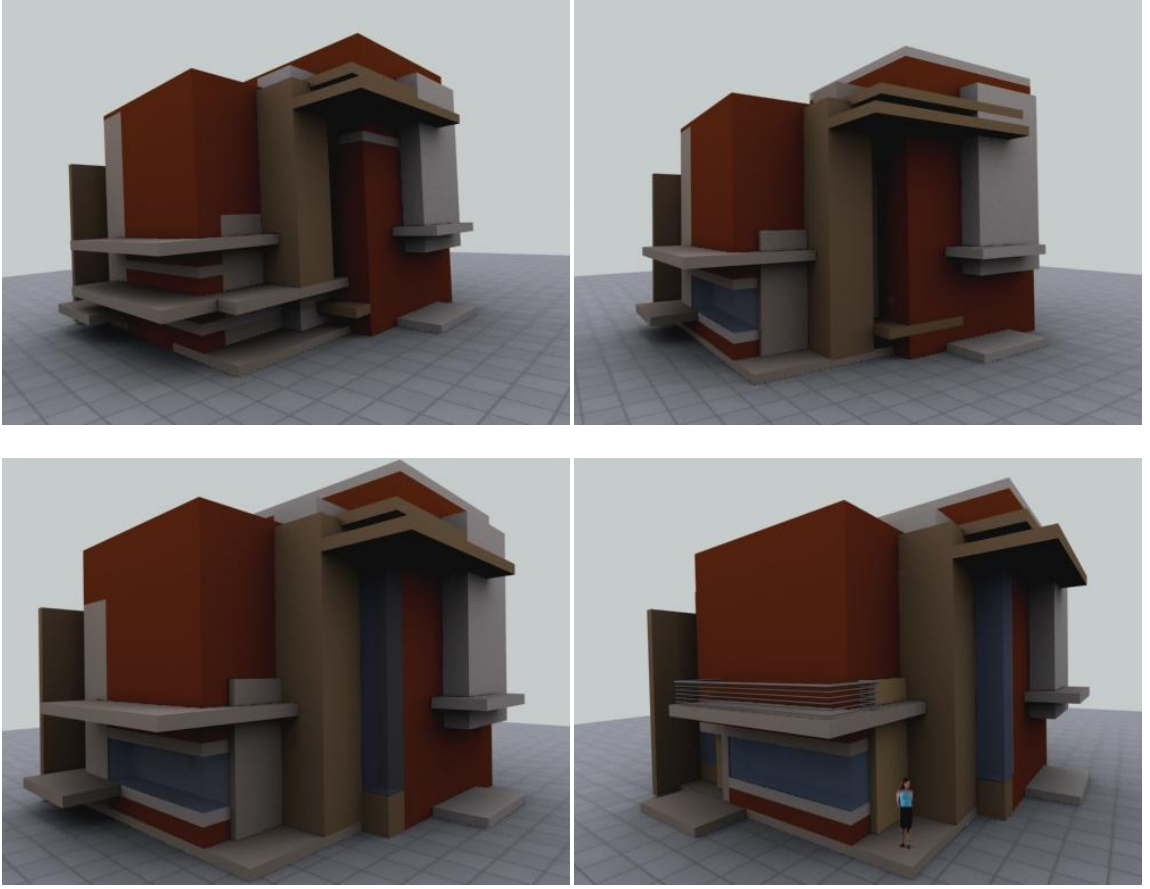
Şekil 4.15 Seçenek 10'a ait görünüşler ve perspektif

Yukarıdaki şekilde seçilen formun cepheleri, üstten görünümü ve perspektifi bulunmaktadır. Seçilen formun az eğimli arazilerde kullanılması hedeflenmiştir. Formun okunması tasarımcının formu algılaması ile ilgilidir. Buna göre, Şekil 4.15'te görüldüğü gibi kat yükseklikleri ve oluşan mekânlar ele alınarak 0 kotu belirlenmiştir.

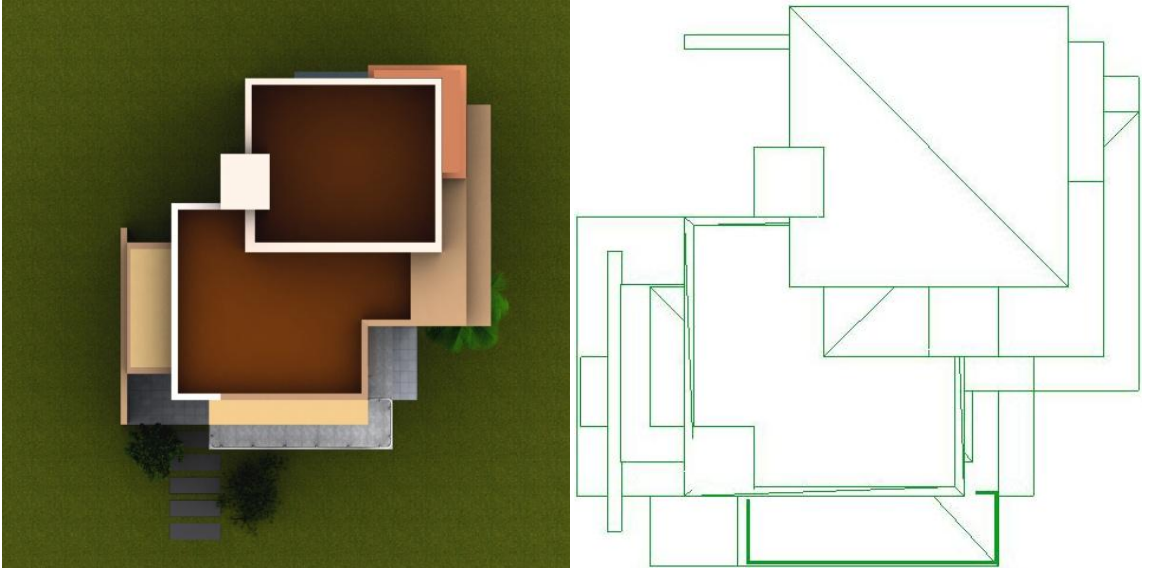


Şekil 4.16 Seçenek 10'a ait perspektif

Tekrarlarla oluşan mekânlar ve boşluklar incelenerek katların kotları ve konutun birimleri ile ilgili çözümler 3 boyutlu düşünülerek tasarım aşamasına geçilmiştir. Şekil 4.17, 4.18, 4.19, 4.20, 4.21, 4.22'de soyut formun konuta dönüşüm süreci görsel olarak ifade edilmiştir. Öncelikle 3 boyutlu olarak düşünülen konut birimlerine göre modele müdahalelerde bulunulmuştur. 3 boyutlu olarak değerlendirilen form, daha sonra ölçüler ve konut birimleri ile ilgili kesin kararların verilmesi için plan düzleminde çalışılmak üzere Archicad programına aktarılmıştır. Archicad programında gerçek ölçeğe uyarlanılarak form üzerinde çalışılmaya devam edilmiş, kat planları çözümlenmiştir. Kat planları çözümlenip konut birimleri belirlendikten sonra konutun cepheleri üzerinde çalışılmıştır. Başlangıç formunda cephelerde görülen hareketler, somut konut projesi tasarımında cepheyi zenginleştirmek için kullanılmıştır.



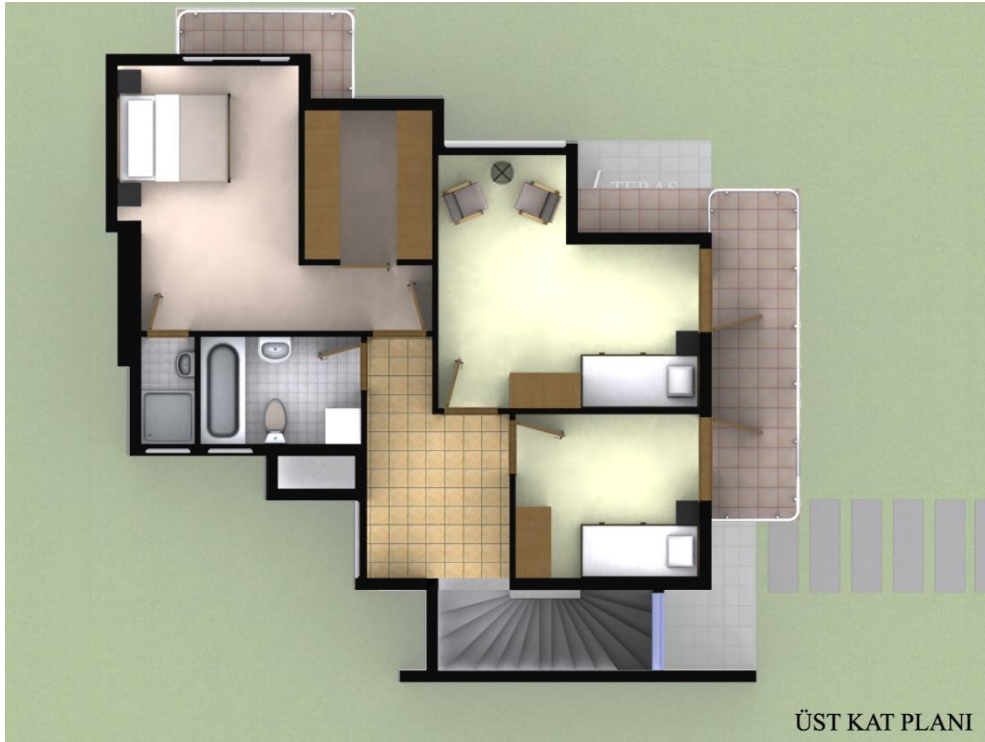
Şekil 4.17 Forma müdahaleler (Seçenek 10)



Şekil 4.18 Üst görünüm (Seçenek 10)



Şekil 4.19 Zemin kat planı (Seçenek 10)



Şekil 4.20 Üst kat planı (Seçenek 10)



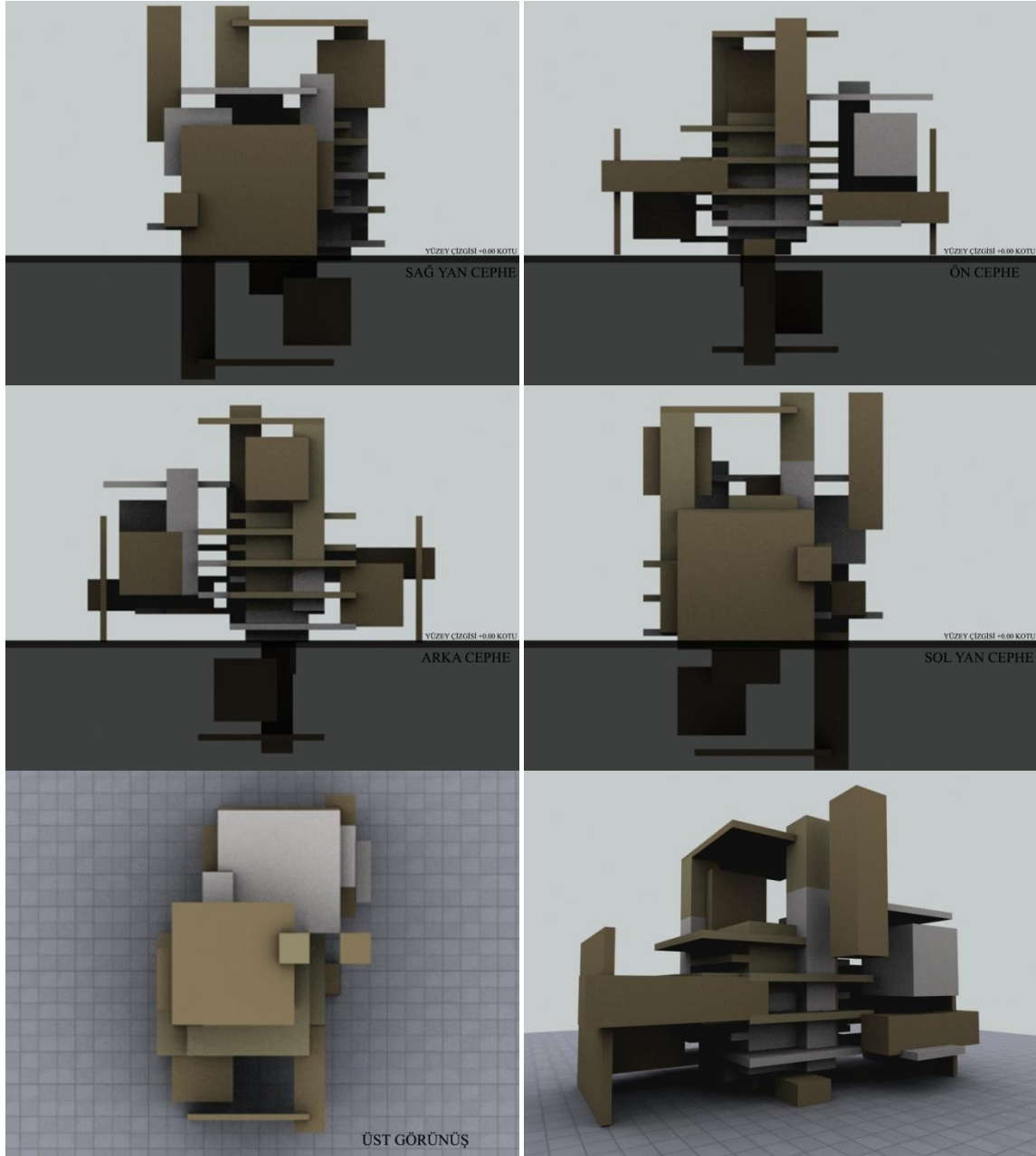
Şekil 4.21 Süreç perspektifleri (Seçenek 10)



Şekil 4.22 Konut perspektifleri (Seçenek 10)

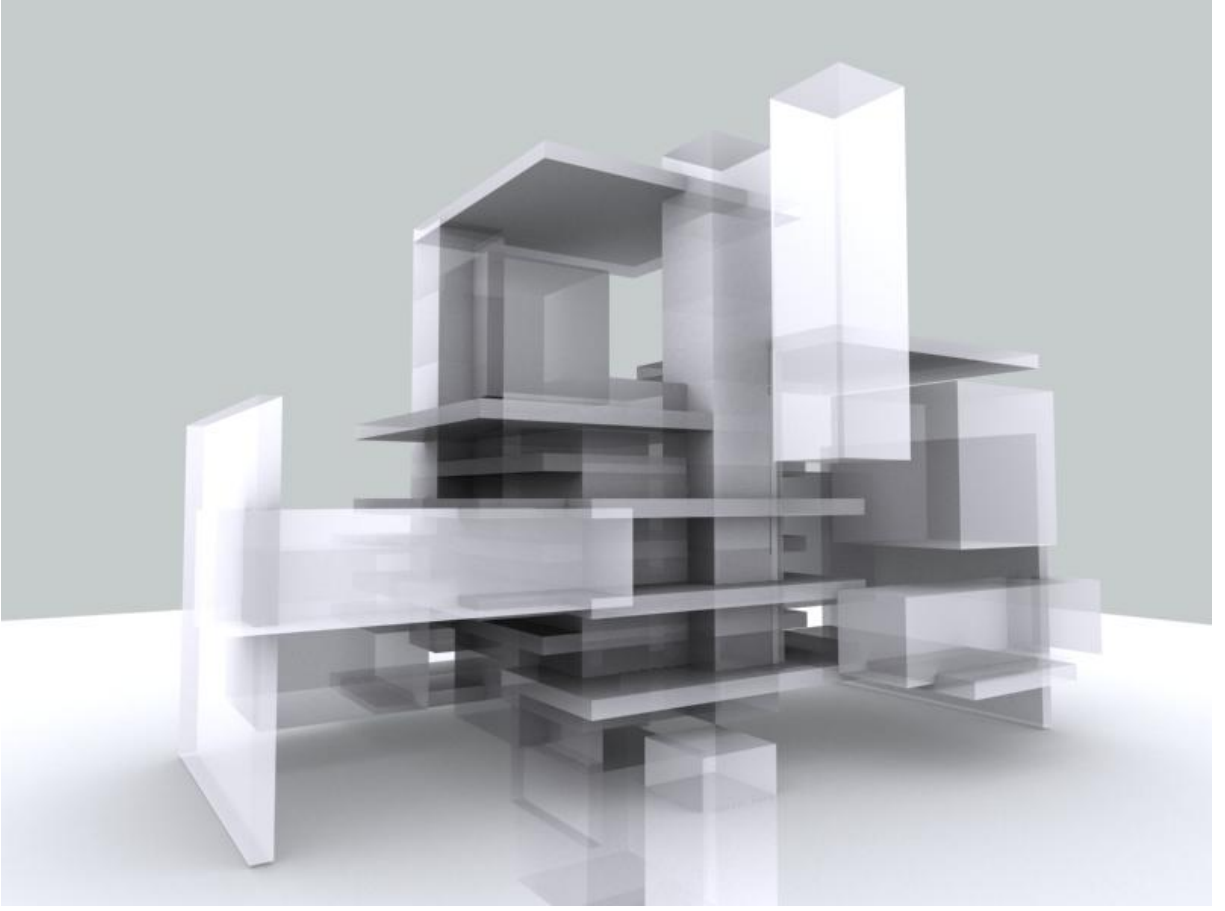


Şekil 4.23 Görünümlerin karşılaştırılması (Seçenek 10)

SEÇENEK 1

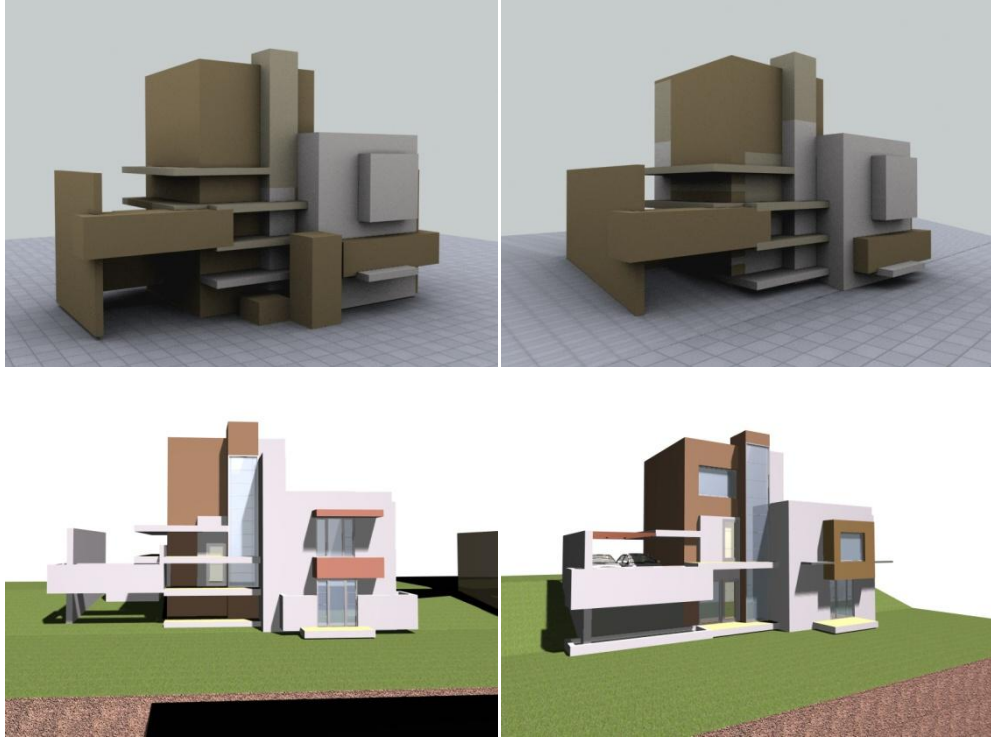
Şekil 4.24 Seçenek 1'e ait görünüşler ve perspektif

Yukarıdaki şekilde seçilen ikinci formun; cepheleri, üstten görünümü ve perspektifi bulunmaktadır. Seçilen formun eğimi fazla olan arazilerde kullanılmasının uygun olacağı düşünülmüştür. Buna göre, Şekil 4.24'te görüldüğü gibi kat yükseklikleri ve oluşan mekânlar ele alınarak O kotu belirlenmiştir.

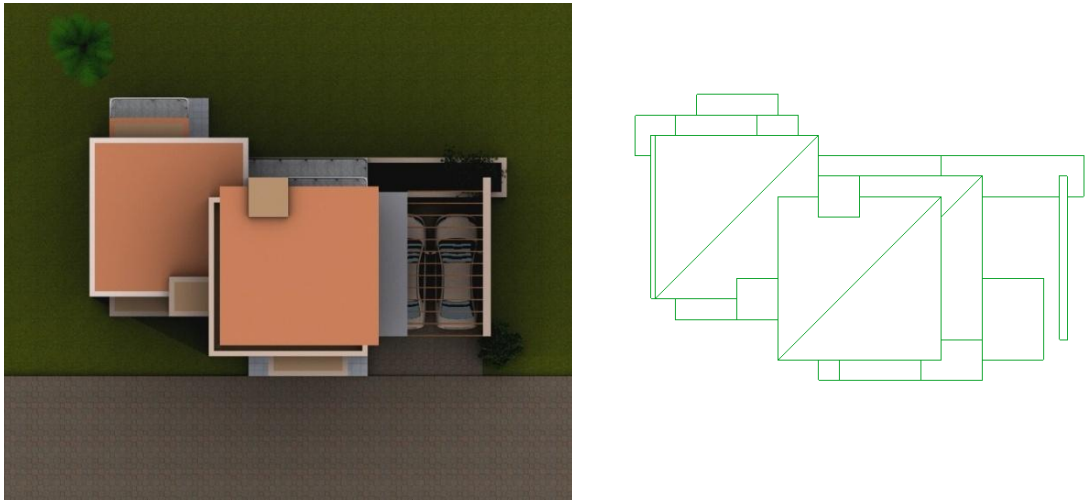


Şekil 4.25 Seçenek 1'e ait perspektif

Form 3 boyutlu olarak tasarlanıp çözümlenirken, kat kotlarının kademeli olması düşünülmüş ve form ona göre yorumlanmıştır. Formun arka cephesi giriş kabul edilmiştir. Arka cephe 2 kat yüksekliğinde olurken arazi eğiminden oluşan kot farkından dolayı ön cephe ise 3 kat yüksekliğindedir. Bu form seçeneğinde otopark olarak tasarlanabilecek bir mekân oluşmuştur. Ayrıca formun kademeli yapısından dolayı tasarlanan kütlelerde teras olarak kullanılacak mekânlar oluşmaktadır. Şekil 4.26, 4.27, 4.28, 4.29, 4.30,4.31'de soyut form kompozisyonunun somut konut projesine dönüşmesi görsel olarak anlatılmıştır. Şekil 4.31'de form türetici ile türetilen formun konut projesi ile karşılaştırılması görsel olarak ifade edilmiştir.



Şekil 4.26 Forma müdahaleler (Seçenek 1)



Şekil 4.27 Üst görüşler (Seçenek 1)

3 boyutlu olarak form üzerinde yapılan müdahaleler ve öngörülen konut birimlerinin plan düzleminde analiz edilmesi için model Archicad programına aktarılmıştır. Form, gerçek ölçeklerde değerlendirilmiş ve plan şemaları çözülmüştür. Oluşan plan şemalarına göre cepheler ve form üzerinde çalışmalar yapılmıştır.



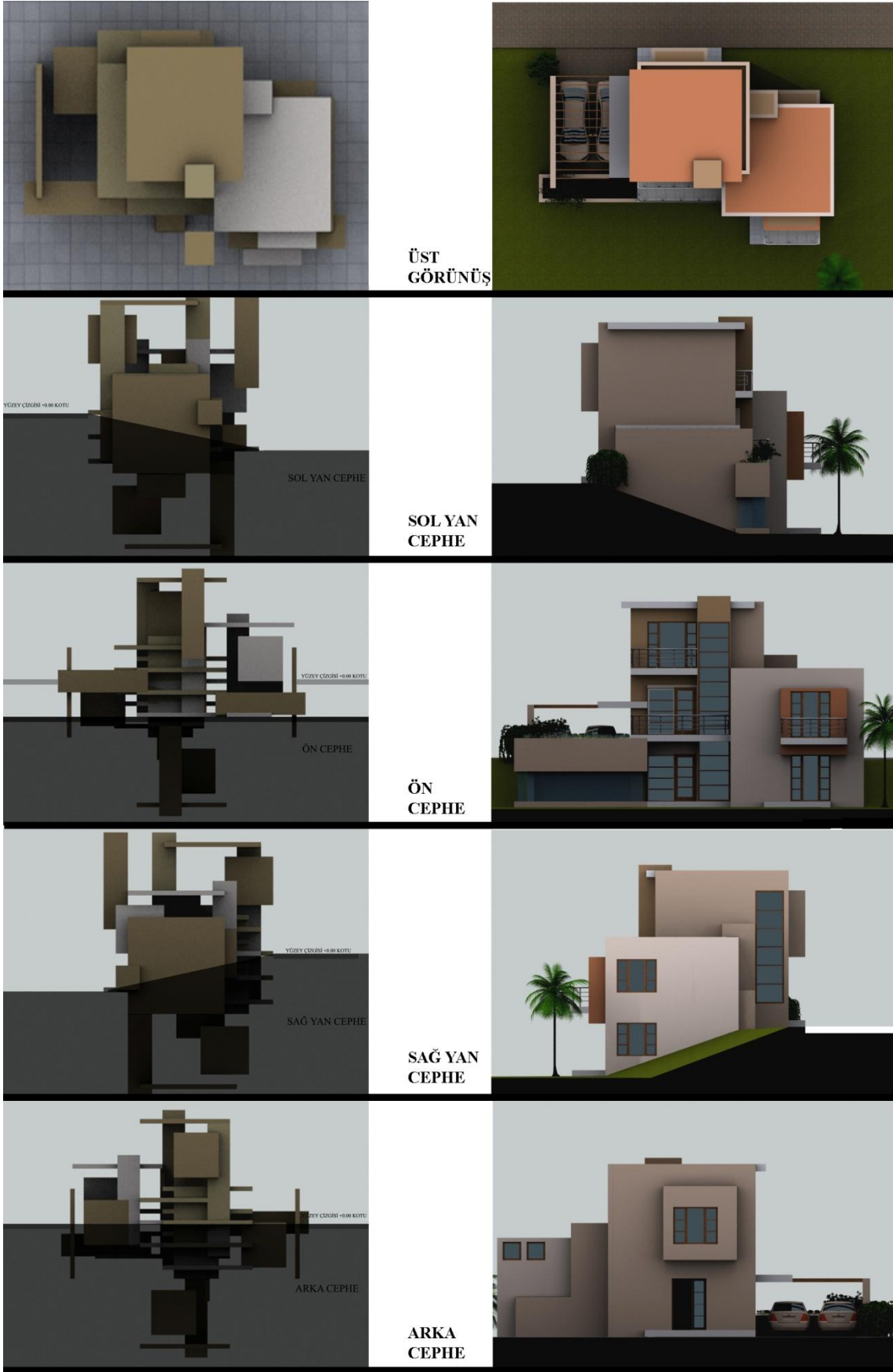
Şekil 4.28 Planlar (Seçenek 1)



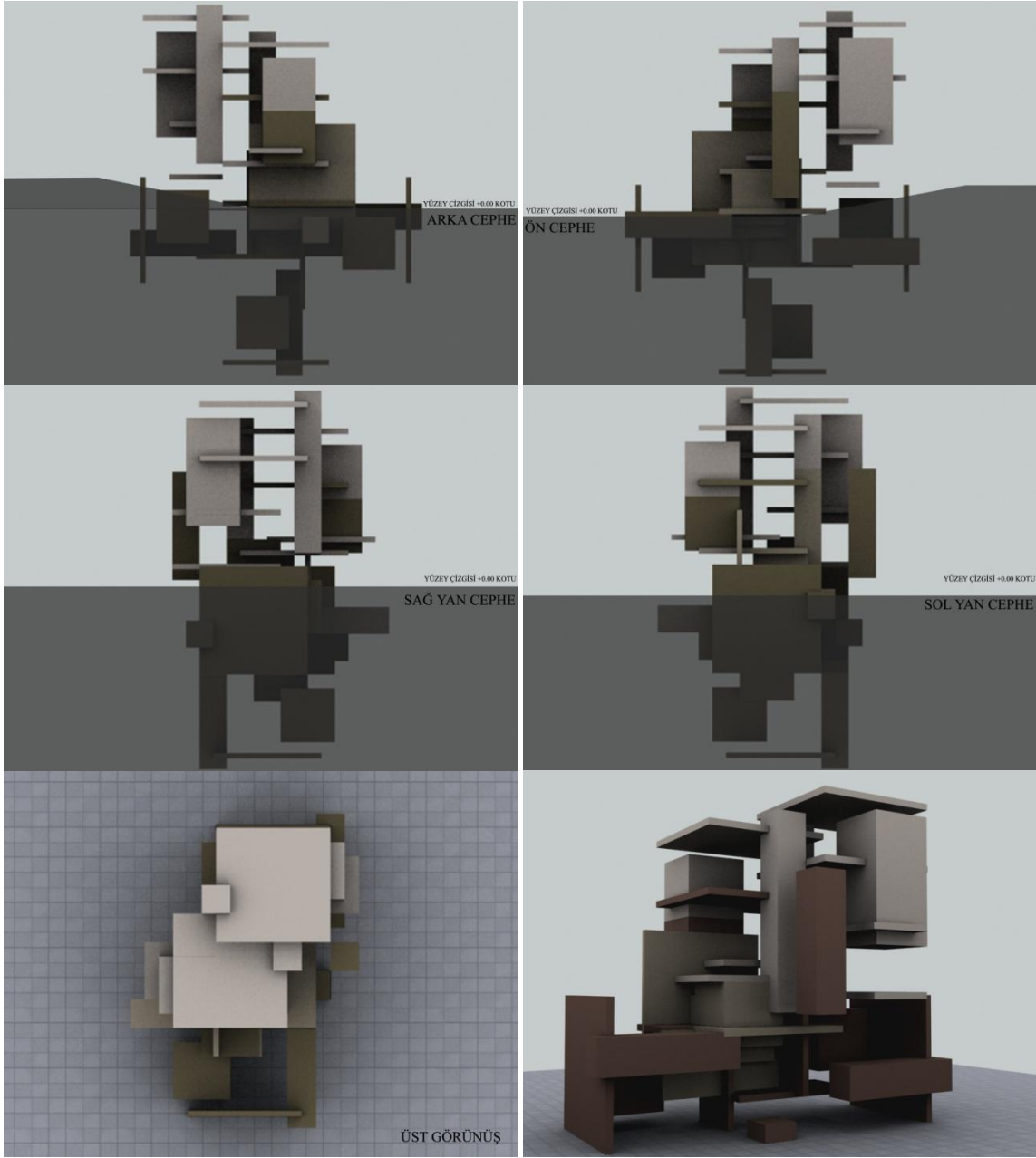
Şekil 4.29 Süreç perspektifleri (Seçenek 1)



Şekil 4.30 Konut perspektifleri (Seçenek 1)

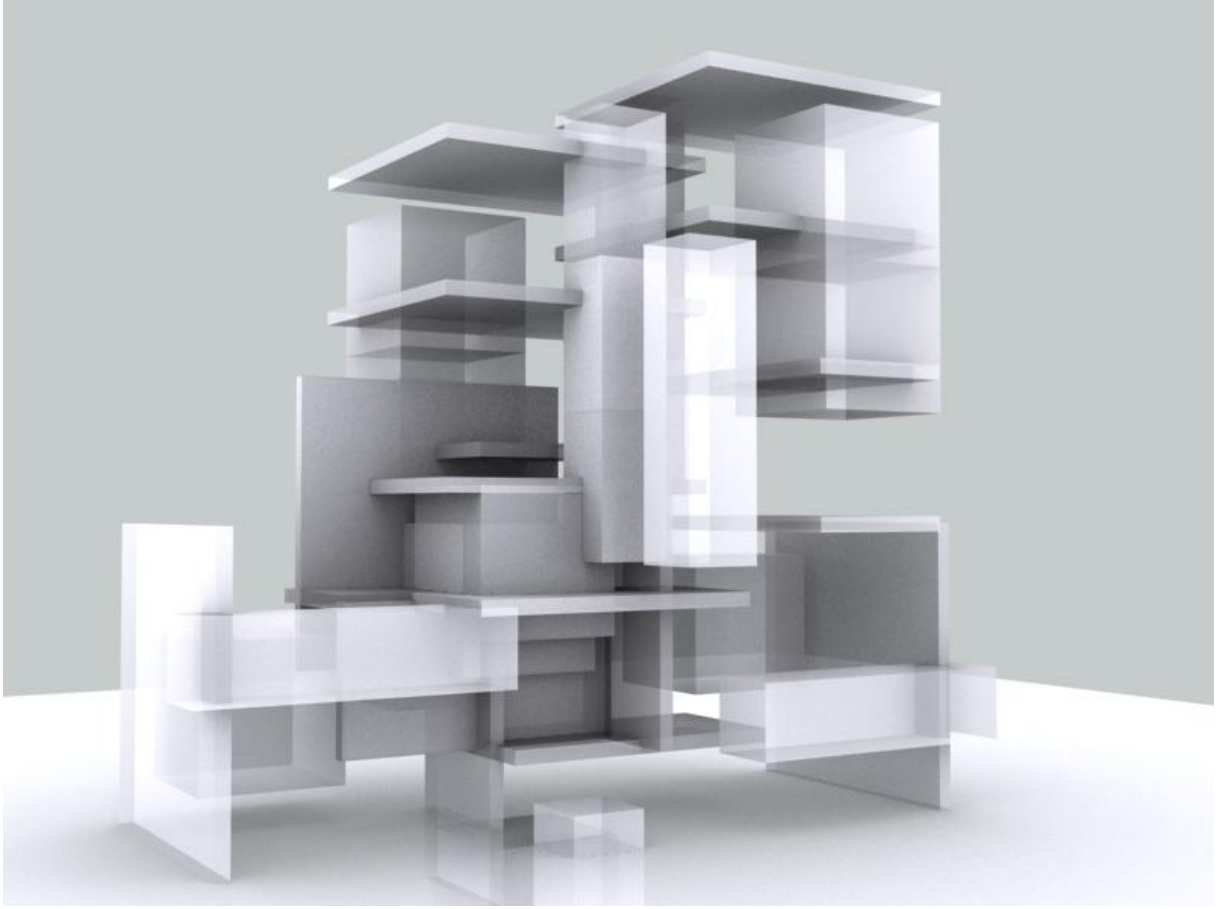


Şekil 4.31 Görünümlerin karşılaştırılması(Seçenek 1)

SEÇENEK 5

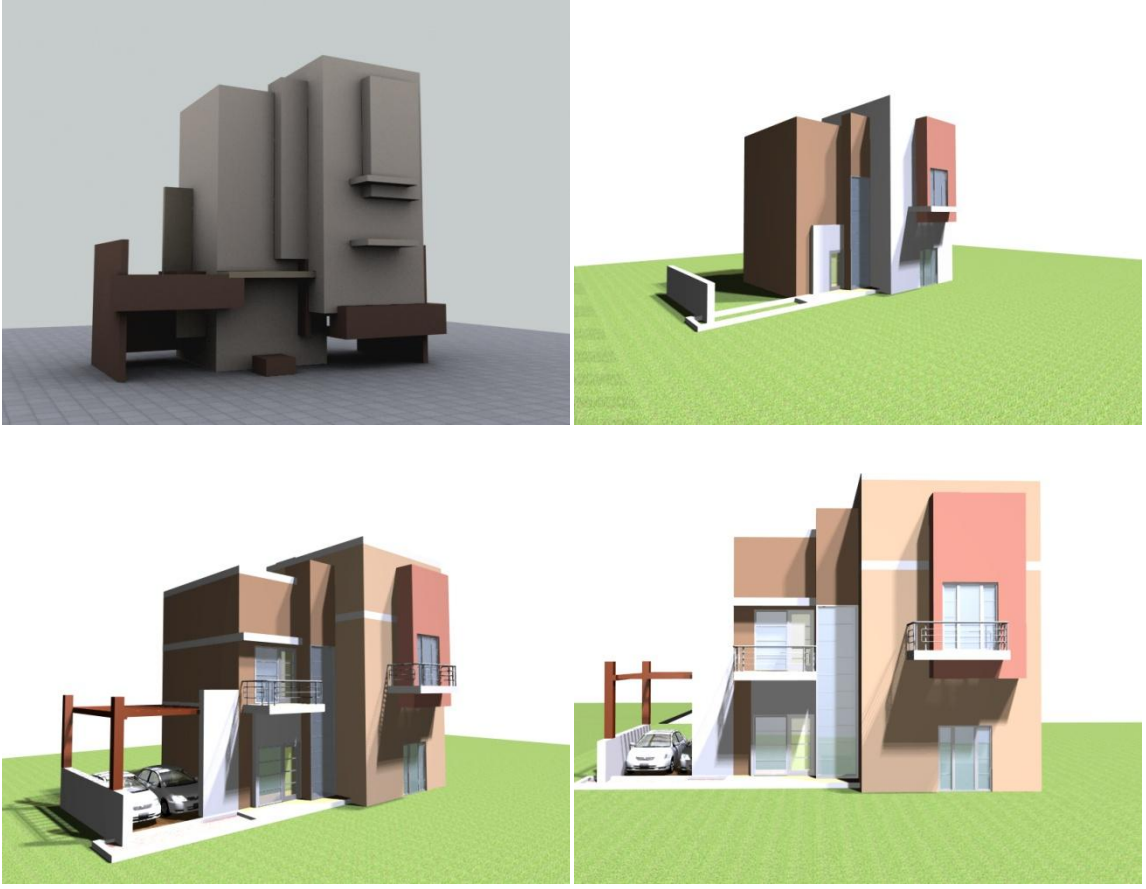
Şekil 4.32 Seçenek 5'e ait görünüşler ve perspektif

Yukarıdaki şekilde seçilen formun cepheleri, üstten görünümü ve perspektifi bulunmaktadır. Seçilen form arazide eğime sağ yan cephesiyle yaslanacak şekilde düşünülmüştür. Buna göre 0 kotu, Şekil 4.32'de görüldüğü gibi, kat yükseklikleri ve oluşan mekânlar ele alınarak belirlenmiştir.



Şekil 4.33 Seçenek 5'e ait perspektif

Oluşan mekânlar ve boşluklar incelenerek katların kotları ve konutun birimleri ile ilgili çözümler 3 boyutlu düşünülerek tasarım aşamasına geçilmiştir. Formun konuta dönüşme sürecinde bilgisayar ortamında yapılan müdahaleler aşağıdaki şekilde aşamalarıyla gösterilmiştir. Öncelikle 3 boyutlu olarak düşünülen konut birimlerine göre modele müdahalelerde bulunulmuştur. 3 boyutlu olarak değerlendirilen form, daha sonra ölçüler ve konut birimleri ile ilgili kesin kararların verilmesi için plan düzleminde çalışılmak üzere Archicad programına aktarılmıştır. Archicad programında gerçek ölçeğe uyarlanılarak form üzerinde çalışılmaya devam edilmiş, kat planları çözümlenmiştir. Kat planları çözümlenip konut birimleri belirlendikten sonra konutun cepheleri üzerinde çalışılmıştır. Başlangıç formundaki cepheler, konut cephesinde çeşitli tasarım öğeleri olarak değerlendirilmiş, tasarıma zenginlik katılmıştır.



Şekil 4.34 Forma müdahaleler (Seçenek 5)



Şekil 4.35 Üst görüşler (Seçenek 5)

3 boyutlu olarak form üzerinde yapılp, öngörülen konut birimlerinin plan düzleminde analiz edilmesi için model Archicad programına aktarılmıştır. Form, gerçek ölçelerde değerlendirilmiş ve plan şemaları çözülmüştür. Oluşan plan şemalarına göre cepheler ve form üzerinde çalışmalar yapılmıştır. Aşağıdaki resimlerde formun konuta dönüşümündeki tasarım süreci ele alınmıştır.



Şekil 4.36 Zemin kat planı (Seçenek 5)



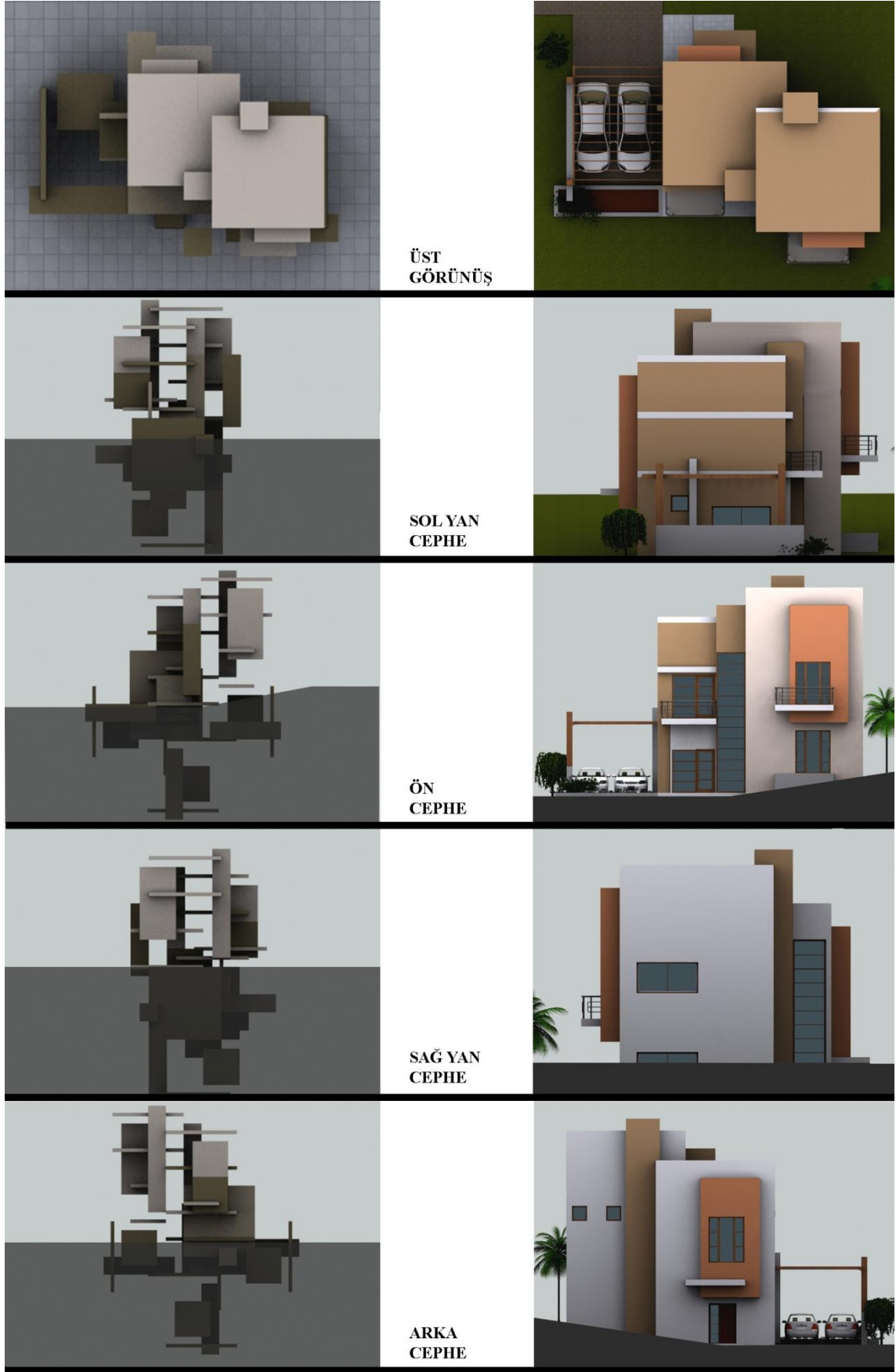
Şekil 4.37 Üst kat planı (Seçenek 5)



Şekil 4.38 Süreç perspektifleri (Seçenek 5)



Şekil 4.39 Konut perspektifleri (Seçenek 5)



Şekil 4.40 Görünüşlerin karşılaştırılması (Seçenek 5)

Arsaya Aplikasyonu

Şekil 4.41 Konut yerleşkesi perspektif

Seçenek 10; A tipi olarak, Seçenek 5; B tipi olarak, Seçenek 1 ise C tipi olarak belirlenmiştir. Konut yerleşkesinde kullanılacak olan villa tiplerinin birbirinden farklı olması amacıyla, 3 villa tipinin de ortak ögesi olan bölüm kavuniçi, turuncu, bordo ve açık sarı ile renklendirilerek 12 adet farklı konut elde edilmiştir. Bu konutlardan 10 tanesi yerleşke içinde kullanılmıştır.



Şekil 4.42 Konut yerleşkesi görünüşleri



Şekil 4.43 Konut yerleşkesi perspektif

Yukarıda konut yerleşme tasarım sürecinde elde ettiğimiz formların araziye aplikasyonu ile elde edilen yerleşkeye ait perspektifler bulunmaktadır. Yerleşke içinde bulunan üç konut tipi birbirileri ile çok farklı ve benzer özellikler taşımaktadır. Aynı biçimlerin ve gramerlerin oluşturduğu formlardan elde edilen konut tipleri aynı ailenin farklı bireyleri olma özelliğini göstermektedirler.

5. SONUÇLAR

Bu çalışma 3 boyutlu Froebel Blok gramerlerinin konut üretiminde bir yöntem olarak kullanılması araştırılmıştır. Bilgisayar ortamında türetmeler yapan Form Türetici, 3ds Max altında çalışan bir plug-in olarak geliştirilmiştir. Froebel form türetici adını verdiğimiz program ile rastgele ya da forma çeşitli aşamalarda müdahaleler de bulunarak yapılan türetmeler, dördüncü bölümde de gösterildiği gibi, mimarın yorumlayıp değerlendirebileceği başarılı sonuçlar verebilmektedir. Froebel form türetici ile herhangi bir mimari tasarım için 3 boyutlu eskizler elde edilebilir. Kullanıcı, türetmeler yaparken süreçleri ve bütün parametreleri kontrol ederek formu istediği şekilde yönlendirebilmektedir. Froebel form türetici ile basit ya da karmaşık sistemler için çözüm aranabilir. 2 blokla,3 blokla veya 4 blokla türetmeler yapılabilmeyle ve 5 dönüşüm noktasına kadar kullanma seçeneği vardır. Her dönüşüm noktasında yeniden tanımlanabilecek parametrelerle zengin bir tasarım uzayı elde edilebilir.

Bu çalışma deneysel bir nitelik taşımaktadır. Gerçek bir uygulamada; çevre, arazi, malzeme, maliyet, strüktür ve kullanıcı talepleri gibi kısıtlar göz önünde bulundurularak soyut formdan somut projeye geçiş tasarımcının yorumuyla gelişecektir. Çalışmada konut yerleşkesi tasarım sürecine etki eden diğer faktörler ve estetik kaygılarla yapılan seçimler yorumlanarak elde edilen konutlar proje kapsamında yerleşke içinde kullanılmıştır.

Froebel form türeticinin birden çok yapıyı içinde barındıran projelerde kullanılması, aynı bloklarla elde edilen farklı formların, birbirileri ile aynı gramerlerden oluşmasından kaynaklanan, aynı ailenin farklı bireyleri olma özelliğini taşımasını sağlamaktadır. Bu sayede elde edilebilecek form zenginliği, mimarlar ve tasarımcıların estetik uyumu yakalama konusundaki kaygılarına çözüm olasılıklarını artırabilecektir.

KAYNAKLAR

Stiny, G. (1980), Introduction to Shape and Shape Grammars, Environment and Planning B: Planning and Design 7:343-351

Stiny, G. (1977), Pictorial and Formal Aspects of Shape and Shape Grammars (Birkhauser, Basel, 1975)

Wang, Y. ve Duarte, J. P. (1998), "Synthesizing 3D forms: shape grammars and rapid prototyping," Workshop on Generative Design, Artificial Intelligence in Design Ö98 Conference, Lisbon, Portugal, , pp. 7-18.

Stiny, G. (1980), "Kindergarten grammars: designing with FroebelÖs building gifts," Environment and Planning B 3: 409-462.

Çolakoğlu, B. (2004) - YTÜ, Fen Bilimleri Enstitüsü, Mimari Tasarım ABD, BOM YL Programı - Biçim Gramerleri Ders Notları

Knight, T. W. (1998), "Designing a Shape Grammar" in J. S. Gero and F. Sudweeks, eds., Artificial Intelligence in Design 98 (Kluwer Academic Publishers), s. 514.

Knight, T. W. "Designing with Grammars," in G. N. Schmitt, ed., Computer-Aided Architectural Design (Verlag Viewag, Weisbaden), pp.33-48.

Duarte, J.P. (2001), Customizing Mass Housing: A Discursive Grammar for Siza's Malagueira Houses - Doktora tezi, Massachusetts Institute of Technology

Knight, W. T. (1994), Shape Grammars and Color Grammars in Design; Environment and Planning B: Vol 21 s 705 - 735

Stiny, G. (1977), Ice-ray: A note on the generation of Chinese lattice designs - Environment and Planning B Vol 4 s 89 - 98

Sass, L. (2001), Reconstructing Palladio's Villas: A Computational Analysis of Palladio's Villa Design and Construction Process, ACADIA Konferansı

Tsui, E. (1999), "Evolutionary Architecture: Nature as a Basis for Design", John Wiley & Son,Inc.

Stiny, G. ve Mitchell, W.J., "The Palladian Grammar" Environment and Planning B 5 (1978): 5-18.

Yazar T., Çolakoğlu B., (2007) - "QSHAPER: A Cad Utility for Shape Grammars",

Stiny, G. (1977), Ice-ray: A note on the generation of Chinese lattice designs - Environment and Planning B Vol 4 s 89 - 98

Stiny, G. ve Knight, W. T. (2001), Classical and Non-classical Computation, Information Technology, Vol 5 no 4 s 355 - 372

Torus, B. (2005), YTÜ, Fen Bilimleri Enstitüsü, Mimari Tasarım ABD, BOM YL Programı - Mimari Proje Atölyesi - Charles Correa Analizleri

Torus, B. (2005) - YTÜ, Fen Bilimleri Enstitüsü, Mimari Tasarım ABD, BOM YL Programı - Mimari Proje Atölyesi – Mardin Mimarisi

İNTERNET KAYNAKLARI

- [1] <http://tr.wikipedia.org/wiki/Algoritma>
- [2] http://www.mmf.gazi.edu.tr/journal/2007_3/DERGI_2007_V22_NO3_pp379-385.pdf
- [3] <http://www.mh-portfolio.com>
- [4] <http://www.metropolistanbul.com/public/temamakale.aspx?mid=21>
- [5] www.mit.edu/%7Etknight/IJDC/frameset_history_design_sg.htm
- [6] http://ddf.mit.edu/papers/01_lsass_acadia_2001.pdf
- [7] http://www.bot.yildiz.edu.tr/_sites/2008_6101/
- [8] http://www.bot.yildiz.edu.tr/_pages/tezCardTR.php?tez=1
- [9] <http://dspace.mit.edu/handle/1721.1/8189>
- [10] <http://www.mit.edu/~4.184/3dshaper.htm>
- [11] http://www.chip.com.tr/arsivpdf/2cf7d69a67c2950f_1999-12-248.pdf

EKLER

Ek 1 Programın 3dsMax programındaki kodundan bir bölüm

Ek 1

```

...

on uret pressed do
if shapeds.state == 1 then
(
    if shapeds.state == 1 then
    (
    A=box()
    A.length=aa
    A.width=bb
    A.height=cc
    A.material=standard diffuse:white opacity:100
    (
    for i=1 to s DO
    (
    B=copy A
    B.length=dd
    B.width=db
    B.height=dy
    in coordsys local move B [xn1,yn1,zn1]
    in coordsys local rotX=eulerangles hh ii jj
    in coordsys local rotate B rotX
    C=copy B
    C.length=aa
    C.width=bb
    C.height=cc
    in coordsys local move C [xn4,yn4,zn4]
    in coordsys local rotX=eulerangles xd4 yd4 zd4
    in coordsys local rotate C rotX
    A=C
    )
    )
    )
    else if shapeds.state == 2 then
    (
    A=box()
    A.length=aa
    A.width=bb
    A.height=cc
    A.material=standard diffuse:white opacity:100
    (
    for i=1 to s DO
    (
    B=copy A
    B.length=dd
    B.width=db
    B.height=dy

```

```

in coordsys local move B [xn1,yn1,zn1]
in coordsys local rotX=eulerangles hh ii jj
in coordsys local rotate B rotX
C=copy B
C.length=kk
C.width=kb
C.height=ky
in coordsys local move C [xn2,yn2,zn2]
in coordsys local rotX=eulerangles oo pp rr
in coordsys local rotate C rotX
D=copy C
D.length=aa
D.width=bb
D.height=cc
D.material=standard diffuse:white opacity:100
in coordsys local move D [xn5,yn5,zn5]
in coordsys local rotY=eulerangles xd5 yd5 zd5
in coordsys local rotate D rotY
A=D
)
)
)
else if shapes.state == 3 then
(
A=box()
A.length=aa
A.width=bb
A.height=cc
A.material=standard diffuse:white opacity:100
(
for i=1 to s DO
(
B=copy A
B.length=dd
B.width=db
B.height=dy
in coordsys local move B [xn1,yn1,zn1]
in coordsys local rotX=eulerangles hh ii jj
in coordsys local rotate B rotX
C=copy B
C.length=kk
C.width=kb
C.height=ky
in coordsys local move C [xn2,yn2,zn2]
in coordsys local rotX=eulerangles oo pp rr
in coordsys local rotate C rotX
D=copy C
D.length=kkk
D.width=kbb
D.height=kyy

```

```

D.material=standard diffuse:white opacity:100
in coordsys local move D [xn3,yn3,zn3]
in coordsys local rotY=eulerangles xd3 yd3 zd3
in coordsys local rotate D rotY
E=copy D
E.length=aa
E.width=bb
E.height=cc
E.material=standard diffuse:white opacity:100
in coordsys local move E [xn6,yn6,zn6]
in coordsys local rotY=eulerangles xd6 yd6 zd6
in coordsys local rotate E rotY
A=E
)
)
)
)   else if shapeds.state == 2 then

(
    if shapeds.state == 1 then

...

```

ÖZGEÇMİŞ

Doğum Tarihi	03.09.1979		
Doğum Yeri	Samsun		
İlkokul	1985-1990	Seyfi Demirsoy İlköğretim Okulu	
Ortaokul-Lise	1990-1997	Samsun Anadolu Lisesi	
Lisans	1998-2004	Karadeniz Teknik Üniversitesi Mühendislik Mimarlık Fakültesi Mimarlık Bölümü	
Yüksek Lisans	2005-2008	Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü Mimarlık Anabilim Dalı Bilgisayar Ortamında Mimarlık Programı	
Çalıştığı Kurumlar	2005	İzgi Mimarlık	(İstanbul)
	2005	İdea İnşaat Ltd. Şti	(İstanbul)
	2006-	Vakıflar Genel Müdürlüğü İstanbul Vakıflar Bölge Müdürlüğü	(İstanbul)