T.C.
YILDIZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

GENİŞ AÇIKLIKLI MEKAN SİSTEMLERİNİN PARAMETRİKLEŞTİRİLMESİNDE BİR YÖNTEM

ENGİN SEDAT ÖZDEMİR

YÜKSEK LİSANS TEZİ
MİMARLIK ANABİLİM DALI
BİLGİSAYAR ORTAMINDA MİMARLIK PROGRAMI

DANIŞMAN
Yrd. Doç. Dr. Togan TONG

İSTANBUL, 2016

T.C.

YILDIZ TEKNİK ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

GENİŞ AÇIKLIKLI MEKAN SİSTEMLERİNİN PARAMETRİKLEŞTİRİLMESİNDE BİR YÖNTEM

Engin Sedat ÖZDEMİR tarafından hazırlanan tez çalışması 26.04.2016 tarihinde aşağıdaki jüri tarafından Yıldız Teknik Üniversitesi Fen Bilimleri Mimarlık Anabilim Dalı'nda **YÜKSEK LİSANS TEZİ** olarak kabul edilmiştir.

Tez Danışmanı

Yrd. Doç. Dr. Togan TONG

Yıldız Teknik Üniversitesi

Jüri Üyeleri

Yrd. Doç. Dr. Togan TONG

Yıldız Teknik Üniversitesi _____

Prof. Dr. Tülin Görgülü

Yıldız Teknik Üniversitesi _____

Yrd. Doç. Dr. Abdurrahman Tuğrul Yazar

Bilgi Üniversitesi _____

ÖNSÖZ

Öncelikle bugünlere gelmemi sağlayan ve her zaman maddi ve manevi desteklerini gördüğüm aileme,

Çalışmam sırasında yüksek sabır göstererek bu tezi tamamlamam için her türlü desteği ve teşviki sunan çok sevgili eşim Nelly ÖZDEMİR'e,

Yüksek lisans eğitimim sırasında birlikte çalıştığım bütün hocalarıma ve arkadaşlarıma,

Lisans ve yüksek lisans öğrenim hayatım boyunca bilgi ve tecrübelerinden faydalandığım, mimarlık ve bilgisayar ilişkisi hakkında bana büyük katkılar sağlayan danışman hocam Sayın Yrd. Doç. Dr. Togan TONG'a,

Sonsuz saygılarımı ve teşekkürlerimi sunarım...

Nisan, 2016

E. Sedat ÖZDEMİR

İÇİNDEKİLER

Sayfa

KISALTMA LİSTESİ vi

ŞEKİL LİSTESİ vii

ÇİZELGE LİSTESİ ix

ÖZET x

ABSTRACT xii

BÖLÜM 1

GİRİŞ 1

1.1 Literatür Özeti 1
1.2 Tezin Amacı 1
1.3 Hipotez 2

BÖLÜM 2

KAVRAMLAR VE TANIMLAR 3

2.1 Geniş Açıklıklı Mekan Sistemleri(GAMS) 3
2.2 GAMS'nin Örneklerle İncelenmesi 5
2.2.1 Kafes-Makas Kiriş Sistemler 5
2.2.2 Kemer Sistemler 7
2.2.3 Uzay Kafes Sistemler 10
2.2.4 Asma Germe Sistemler 13
2.2.5 Pnömatik Sistemler 15
2.3 Parametrik Tasarım Kavramı 17
2.3.1 Parametrik Modelleme 19
2.3.2 Mimaride Kullanılan CAD Programları 22
2.3.2.1 AutoCAD 24
2.3.2.2 Allplan 25
2.3.2.3 ArchiCAD 27
2.3.2.4 Revit 28
2.3.2.5 Generative Components 30

2.3.2.6 Lumion 31
2.3.2.7 Unity 32
2.3.2.8 Unreal Engine 33
2.3.2.9 Maya (MelScript) 34
2.3.2.10 Rhinoceros (RhinoScript - Grasshopper) 36
2.3.2.11 3DS Max (MaxScript) 39
2.4 MaxScript Kavramı 17

BÖLÜM 3

PARAMETRİKLEŞTİRİLMİŞ STRÜKTÜR SİSTEMLERİ (PSS) 47

3.1 GAMS ve MaxScript İlişkisinin Kurulması 47
3.2 PSS (Parametrikleştirilmiş Strüktür Sistemleri) Programı 50
3.3 PSS Programı Kullanımı 55
3.4 PSS Programı ile Yapılmış Örnekler 66

BÖLÜM 4

SONUÇ VE ÖNERİLER 70

KAYNAKLAR 72

EK-A

Kafes Sistem SAP2000 Çözümleri 74

A-1 Howe Tipi Paralel Başlık Kafes Sistem SAP2000 Çözümü 74
A-2 Howe Tipi Üçgen Başlık Kafes Sistem SAP2000 Çözümü 75
A-3 Pratt Tipi Paralel Başlık Kafes Sistem SAP2000 Çözümü 76
A-4 Pratt Tipi Üçgen Başlık Kafes Sistem SAP2000 Çözümü 77
A-5 Warren Tipi Paralel Başlık Kafes Sistem SAP2000 Çözümü 78
A-6 Warren Tipi Üçgen Başlık Kafes Sistem SAP2000 Çözümü 79

EK-B

PSS Programı MaxScript Kodları 80

ÖZGEÇMİŞ 105

KISALTMA LİSTESİ

BIM	Yapı Bilgi Sistemi (Building Information Modeling)
CAD	Bilgisayar Destekli Tasarım (Computer Aided Design)
GAMS	Geniş Açıklıklı Mekan Sistemleri
PSS	Parametrikleştirilmiş Strüktür Sistemleri (Programı)
SCRIPTING	Kod Dizisi Oluşturma

ŞEKİL LİSTESİ

Sayfa

Şekil 2.1 Kafes kiriş tiplerinden bir kaç kesit örneği....6
Şekil 2.2 Bir kafes kiriş örneği (parabolik-paralel başlıklı)....7
Şekil 2.3 Kemer kiriş sistem kesit şeması....8
Şekil 2.4 Kemer kiriş sistem örneği – Atina Olimpiyat Stadı Girişi....9
Şekil 2.5 Uzay kafes sistem örnek detay....11
Şekil 2.6 Uzay kafes sistemle oluşturulmuş örnek form....12
Şekil 2.7 Kablo system....13
Şekil 2.8 Asma germe sistem örneği – Millenium Dome....14
Şekil 2.9 Çift cidarlı pnömatik sistem kesit şeması....15
Şekil 2.10 Çift cidarlı-kanallı pnömatik sistem örneği – BC Place Stadium....16
Şekil 2.11 Bir dikdörtgen ile bir çember arasındaki ilişkisel transformasyonları gösteren bir örnek....21
Şekil 2.12 Sydney Opera Evi'nin parametrik modellemesi....22
Şekil 2.13 AutoCAD Ekran Görüntüsü....25
Şekil 2.14 AllPlan Ekran Görüntüsü....26
Şekil 2.15 ArchiCAD Ekran Görüntüsü....28
Şekil 2.16 Revit Architecture 2016 Ekran Görüntüsü....29
Şekil 2.17 Generative Components Ekran Görüntüsü....31
Şekil 2.18 Lumion Ekran Görüntüsü....32
Şekil 2.19 Unity Ekran Görüntüsü....33
Şekil 2.20 Unreal Engine Ekran Görüntüsü....34
Şekil 2.21 Maya 2016 Ekran Görüntüsü....35
Şekil 2.22 Maya Mel-Script Ekran Görüntüsü....36
Şekil 2.23 Rhinoceros Ekran Görüntüsü....37
Şekil 2.24 RhinoScript Ekran Görüntüsü....38

Şekil 2.25 Grasshopper Ekran Görüntüsü 39
Şekil 2.26 3DS Max 2016 Ekran Görüntüsü 41
Şekil 2.27 MaxScript Kullanılmakta Olan Bir 3DS Max Ekran Görüntüsü 42
Şekil 2.28 MaxScript ekran görüntüsü 45
Şekil 3.1 Howe Tipi paralel başlık kafes sistem SAP2000 çözümü.....48
Şekil 3.2 Howe tipi üçgen başlık kafes sistem SAP2000 çözümü 49
Şekil 3.3 P.S.S. Programı arayüz ve kod dizisi ekran görüntüsü 51
Şekil 3.4 P.S.S. Programı hata ekranı – kiriş yüksekliği yetersiz olduğunda 52
Şekil 3.5 P.S.S. Programı hata ekranı – kolonlar arası mesafe çok az veya çok fazla olduğunda 52
Şekil 3.6 P.S.S. Programı kurulumunu anlatan ekran görüntüleri.....56
Şekil 3.7 P.S.S. Programı arayüzü ve açıklaması.....57
Şekil 3.8 P.S.S. Programı arayüzü – kafes sistemler menüsüne geçiş ve açıklaması.....58
Şekil 3.9 P.S.S. Programı arayüzü – paralel başlık çeşitleri ve arayüz bölüm açıklamaları 59
Şekil 3.10 P.S.S. Programı Arayüzü – Yardımcı Pop-upların Toplu Görüntüsü.....60
Şekil 3.11 P.S.S. Programı Arayüzü – Parametrik Veri Penceresi Açıklamarı 61
Şekil 3.12 P.S.S. Programı Arayüzü – Üçgen Başlıklı-Howe Penceresine Geçiş 62
Şekil 3.13 P.S.S. Programı Arayüzü – Trapez ve Parabolik Başlık Çeşitleri 63
Şekil 3.14 Programı paralel başlıklı kafes kiriş sistemler akış diyagramı.....64
Şekil 3.15 P.S.S. Programı üçgen başlıklı kafes kiriş sistemler akış diyagramı 65
Şekil 3.16 Üçgen Başlıklı Howe Kafes Kiriş Görüntüsü – Dış Perspektif 66
Şekil 3.17 Üçgen Başlıklı Howe Kafes Kiriş Görüntüsü – Dış Perspektif 66
Şekil 3.18 Paralel Başlıklı Howe Kafes Kiriş Görüntüsü – Dış Perspektif 67
Şekil 3.19 Paralel Başlıklı Howe Kafes Kiriş Görüntüsü – İç Perspektif 67
Şekil 3.20 Paralel Başlıklı Warren-Simple Kafes Kiriş Görüntüsü – Dış Perspektif 68
Şekil 3.21 Paralel Başlıklı Warren-Simple Kafes Kiriş Görüntüsü – İç Perspektif 68
Şekil 3.22 Paralel Başlıklı Warren-Modified Kafes Kiriş Görüntüsü – Dış Perspektif 69
Şekil 3.23 Paralel Başlıklı Warren-Modified Kafes Kiriş Görüntüsü – İç Perspektif 69

ÇİZELGE LİSTESİ

Sayfa

Çizelge 2. 1 İncelenen Programlar ve İşlev Tablosu 43
Çizelge 2. 2 Export edilen modelin incelenen programlarda hangi formatların desteklendiğini gösteren çizelge 46
Çizelge 3 1 Geçilen açıklık ve sistem uzunluğu değişkenleri sonuç çizelgesi (6 aks üzerinden oluşturulmuştur.) 54

ÖZET

GENİŞ AÇIKLIKLI MEKAN SİSTEMLERİNİN PARAMETRİKLEŞTİRİLMESİNDE BİR YÖNTEM

Engin Sedat ÖZDEMİR

Mimarlık Anabilim Dalı

Yüksek Lisans Tezi

Tez Danışmanı: Yrd. Doç. Dr. Togan TONG

Değişen mimari yaklaşımların ve gelişen teknolojinin ışığında günümüz mimarlığı ciddi değişimler göstermektedir. Bilgisayarın mimarlığın içine ciddi şekilde girmeye başlamasıyla bu süreç hızlanmış ve sürekli olarak kendisini yenilemiştir. Dolayısı ile bizlerin bu gelişimin içinde olmamız yadsınamaz bir durumdur. Bu çalışma için; birbirinden farklı 3 konunun araştırılması ve bunların ortak bir noktada birleştirilerek sonuç ürüne gidilmesi şeklinde söylenebilir.

İlk bölümlerde çalışmanın izlediği yol ve amaçlardan bahsedilmiştir. Kimler için ve nasıl kullanılabileceği anlatılmıştır.

İkinci bölüm içerisinde geniş açıklıklı mekan sistemleri(GAMS), parametrik tasarım ve scripting konuları incelenmiş ve tanımları yapılmıştır. GAMS için oluşturulmuş sınıflandırlanmalardan bahsedilerek, Heino Engel'in sınıflandırmaları anlatılmıştır. Parametrik tasarım için, temel sözlük anlamından başlayarak parametrik geometrilerden ve modellemelerden bahsedilmiş ve parametrik alt yapıya sahip mimari programlar incelenmiştir. Tüm bu programların avantajlarından ve dezavantajlarından bahsedilmiştir. Scripting konusuna en uygun programın 3DS Max altında çalışan MaxScript olduğuna kararverilmiş; devamında ise Scripting konusu ışığında MaxScript incelenmiştir.

Üçüncü bölümde, GAMS'ye örnekler verilerek kafes sistemler, kemer sistemler, uzay-kafes sistemler, asma germe sistemler ve pnömatik sistemler anlatılmıştır. GAMS'nin parametrik yapısından bahsedilmiş ve MaxScript ile ilişkisi anlatılmıştır. Burada sonuç

ürün hakkından bilgiler verilmeye başlanmıştır. Parametrik Strüktür Sistemleri (PSS) olarak tanımladığım programın içeriği, nerede ve hangi amaçla kullanılabileceği anlatılmıştır. Basit bir arayüzle hazırlanmışta olsa son olarak PSS programının nasıl kullanılacağı resimli örneklerle anlatılmıştır.

Son bölümde, yani sonuç kısmında, parametrik tasarım, GAMS ve Scripting konularının bütünlüğü anlatılmıştır. Bilgisayar Ortamında Mimarlık Programı'nda Scripting örneği olarak faydalanılabileceğinden ve aynı zamanda Yapı Programı'nda da öğrencilerin GAMS'yi daha kolay anlayabileceği bir program olarak kullanılabileceğinden bahsedilmiştir. Yapılan MaxScript kodları ekler kısmında verilmiştir.

Anahtar kelimeler: Parametrik Tasarım, Scripting, Geniş Açıklıklı Mekan Sistemleri, Bilgisayar Destekli Tasarım, Parametrik Programlar

ABSTRACT

A METHOD FOR MAKİNG PARAMETRİC WİDE SPAN SYSTEMS

Engin Sedat ÖZDEMİR

Department of Architecture

MSc. Thesis

Adviser: Asst. Prof. Dr. Togan TONG

Present architecture has serious changes in the light of incremental architectural approaches and developing (progressive) technology. This process has accelerated and has renewed itself through starting to use computers in architecture. Therefore, it is undeniable to be included in this development as architect. We can summarize this study, as of searching 3 different subjects (topics) and to combine these on the common ground and getting result.

On the first chapter, we talked about the way of study and aims. We explained who and how can be used it.

On the second chapter, wide span structure systems, parametric design, scripting topics analyzed and described. Talking about composed classification for wide span structure systems, we explained Heino Engel classifications. For the parametric design, we mentioned parametric geometry and modeling, beginning basic dictionary meaning and parametric architectural programs has been analyzed. Advantages and disadvantages of these programs has been mentioned. The most convenient program for the scripting is MaxScrip that working under 3DS Max has been decided. And MaxScript has been analyzed for the scripting.

On the third chapter, exampling for wide span structure systems (GAMS), truss systems, arch systems, space frame systems, cable and tension systems ve pneumatic systems has been explained. Talked about the parametric structure of wide span structure systems (GAMS) and explained the relation with MaxScript. At here, started informing about result. The content of the program that has been described parametric structure

systems (PSS) by me, explained where and how can be used. Finally, how can be used PSS program has been explained with picture samples even it has a simple interface.

On the last chapter that it is result chapter, the integrity of parametric design, wide span structure systems (GAMS) and scripting topics has been explained. Talked about the program, Both in the Bilgisayar Ortamında Mimarlık Program can be used as scripting examples and in the Yapı Program, the student of GAMS can be understood and can be used easily. The MaxScript codes has been added.

Keywords: Parametric Design, Scripting, Wide Span Structure Systems, Computer Aided Design, Parametric Programs.

BÖLÜM 1

GİRİŞ

1.1 Literatür Özeti

Araştırma sürecinde,

-Geniş açıklıklı mekan sistemlerinin(GAMS) araştırılmıştır;

-Parametrik tasarım üzerine literatür araştırması yapılmıştır;

-GAMS'nin parametrik alt yapısının incelenmiştir;

-Parametrik programların araştırılıp incelenlemiştir;

-Scripting üzerine çalışmalar yapılması ve örnekler incelenmiştir;

-Parametrik programlar ve scripting bütünlüğünün incelenmiştir;

-Sonuç ürüne ulaşmak için ideal program seçimi yapılmıştır;

-İdeal program için gerekli alt yapı bilgisinin araştırılmıştır;

-GAMS için gerekli verilerin programa girilmiştir;

-Oluşturulması hedeflenen sonuç ürünün kodlamalarının yapılmıştır;

-Sonuç ürün için testler yapılmıştır.

1.2 Tezin Amacı

Geniş açıklıklı mekan sistemleri(GAMS), parametrik tasarım ve scripting konuları araştırılarak, scripting mantığına sahip bir program ile üniversitemizin Bilgisayar Ortamında Mimarlık ve Yapı Programlarında eğitim ve araştırma amaçlı kullanılabilecek bir kod grubu üretmektir. Bu kod grubunun kendi arayüzü var ve bu arayüz ile kullanıcı,

uzun sürecek olan işlerini sadece birkaç tıklama ile gerçekleştirebilecektir. Aynı zamanda değişiklik imkanları sağlayacak ve proje tasarım aşamalarında kullanıcıya yön verebilmektir.

1.3 Hipotez

Tez çalışmam iki farklı aşamanın birleştirilmesinden oluşmuştur. Bunlardan ilki daha önce oluşturulmuş geniş açıklıklı mekan sistemlerinin araştırılmasıyla ve benzer sistemlerle veya yeni sistemlerle oluşturulabilecek mekan sistemlerinin önerilmesiyle olmuştur. İkinci aşama ise tüm bu araştırmaların ve önerilerin 3DS Max programı altında MaxScript kod sisteminin kullanılarak parametrik olarak hazırlanmasıyla meydana gelmiştir.

Temel geniş açıklıklı mekan sistem grupları esas alınarak yeni gruplandırmalar oluşturulmuş ve MaxScript uygulama aşaması için teknik bilgiler (kod grupları ve temel görselleştirmeler) haline getirilmiştir. Tüm bu kodlamalar, oluşturulmuş olan bu sistemlerin ön statik hesaplarının yapılması ile oluşturulmuştur. Bu hesaplamalar inşaat ve statik firmalarından destek alınarak sonuca gidilmiştir.

3DS Max programı altında çalışan MaxScript kodlarıyla bir kod grubu üretilmiş ve 3DS Max programında bir program yaması şeklinde çalışmaktadır. Kendisine ait bir arayüzü vardır ve bu arayüzden parametrik olarak oluşturulması planlanan sistemlere erişilerek ve veriler girilerek sistemin modeli oluşturulabilmektir. Bu aşamada yapılan çalışma yeni sistem tasarımlarına açık bir şekilde düşünülmüştür. Daha önce yapılmış olan statik hesaplamalar MaxScript mantığına uygun bir şekilde matematiksel veriler olarak kod sistemine işlenmiştir ve programa parametreleri girilerek oluşturulacak olan sistemin diğer temel statik bilgileri de eş zamanlı olarak kullanıcıya gösterilmesi hedeflenmiştir.

BÖLÜM 2

KAVRAM VE TANIMLAR

2.1 Geniş Açıklıklık Mekan Sistemleri (GAMS)

18. yy'dan başlayarak günümüze kadar uzanan zaman dilimi içerisinde toplumsal ve teknolojik gelişmelere paralel olarak yapıların strüktürlerinde de gelişmeler olmuştur. 20. yy'dan itibaren yüksek fırınların çelik üretiminde kullanılmaya başlanmasıyla, üretim artmış ve modern çelik yapı tekniğinde büyük ilerlemeler kaydedilmiştir [1].

Çelik yapılardaki gelişmenin ikinci aşaması, kaynaklı birleşimlerin yapısal alanda kullanılmaya başlanmasıyla gerçekleştirilmiştir. Kaynaklı birleşimlerin ilk önce yüksek yapılarda daha sonraları köprülerde kullanılmaya başlanmasıyla birlikte bugünkü modern konstrüksiyon anlayışı ortaya çıkmıştır [1].

Metalürji alanında yapılan bilimsel araştırmaların sonucunda, günümüzde yapı alanında kullanılan yüksek dayanımlı çelikler bulunmuştur. Bu malzeme kullanılarak, açıklıkları 1.000m'yi geçen çelik asma köprüler inşa edilebilmiştir. Çelik malzemedeki teknolojik ilerleme ile birlikte çeliğin üstün özellikleri ortaya çıkmıştır. Malzemedeki ilerleme, büyük açıklıklı çağdaş çelik strüktürel sistemlerin gelişmesini hızlandırmıştır. Bu sistemler, geleneksel yapım sistemlerinden malzeme, yük taşıma ilkeleri, statik ilkeler vb. yönlerden farklı özellikler gösteren, daha büyük açıklıkları örtmeye uygun sistemlerdir [1].

Yapı ve malzeme teknolojisindeki ilerlemeler ile birlikte, çeşitli strüktürel sistemler gelişmiştir. Bu taşıyıcı sistemler araştırmacılar tarafından incelenmiş ve bu sistemler farklı sınıflandırma grupları içerisinde toplanmıştır. Strüktürel sistemlerin

sınıflandırılmaları bu sistemlerin anlaşılmasını, özelliklerinin daha iyi kavranabilmesini ve taşıyıcı sistem seçiminin daha uygun şekilde yapılmasını sağlamıştır.

Büyük açıklıklı çelik taşıyıcı sistemlerin mimarinin gelişmesinde ve yapılardaki strüktürel sınırların aşılmasında önemli etkileri olmuştur. Bu nedenle bu sistemlerin araştırılmalarına önem verilmesi gerekmektedir.

İnsanların kendilerini ve varlıklarını dış etkilerden korumak ve çeiştli amaçlarla kullanmak üzere inşa ettikleri yapılar, türüne göre farklı taşıyıcı sistemler gerektirirler. Birbirlerine oldukça karmaşık biçimlerde kaynaştıkları için taşıyıcı sistemleri sınıflamak pek kolay değildir. Bununla beraber en belirgin görünen özelliklerini belirlemek süretiyle taşıyıcı sistemler çeşitli bakımlardan sınıflandırılmışlardır.

-İlk çağlardan itibaren yapıların Taşıyıcı Sistemlerinin tarihsel gelişimi göz önüne alınarak yapılan sınıflandırma : (Geleneksel Taşıyıcı Sistemler, Çağdaş Taşıyıcı Sistemler).

-Yapıların Rijitliklerine göre sınıflandırma : (Rijit Yapılar, Esnek Yapılar).

-Taşıyıcı elemanların boyutlarına bağlı olarak yapılan sınıflandırma: (Bir Boyutlu Yapılar-Ankastre konsol kirişler gibi, İki Boyutlu Yapılar-Düzlem çerçeveler gibi, Üç Boyutlu Yapılar-Uzay kafes sistemler gibi).

-Taşıyıcı sistem türlerine göre sınıflandırma: (Kolon-kiriş sistemi, Çekmeye ve Basınca Çalışan Sistemler, Kirişler, Çerçeveler ve Kemerler, Izgara Sistemler, Plaklar ve Katlanmış Plaklar, Kabuklar, Membranlar, Şişirme(Pnömatik) Sistemler)

Bunlardan bir diğeri ise, mimar mühendis ve bilim adamı olan Frei OTTO'nun Katı, Akışkan ve Plastik gibi çok farklı ana başlıklar altında sınıflandırmasını görüyoruz [2].

Bir diğer sınıflandırma türü olan ve benim tezimin çıkış noktasını oluşturan Heino Engel'in "Structure Systems" kitabında yer alan sınıflandırma metodu.

- Aktif Form Strüktür Sistemler
- Aktif Vektör Strüktür Sistemler
- Aktif kesit strüktür sistemleri
- Aktif Yüzey Strüktür Sistemleri
- Karma Strüktür Sistemleri

- Geometri ve Strüktür Formu

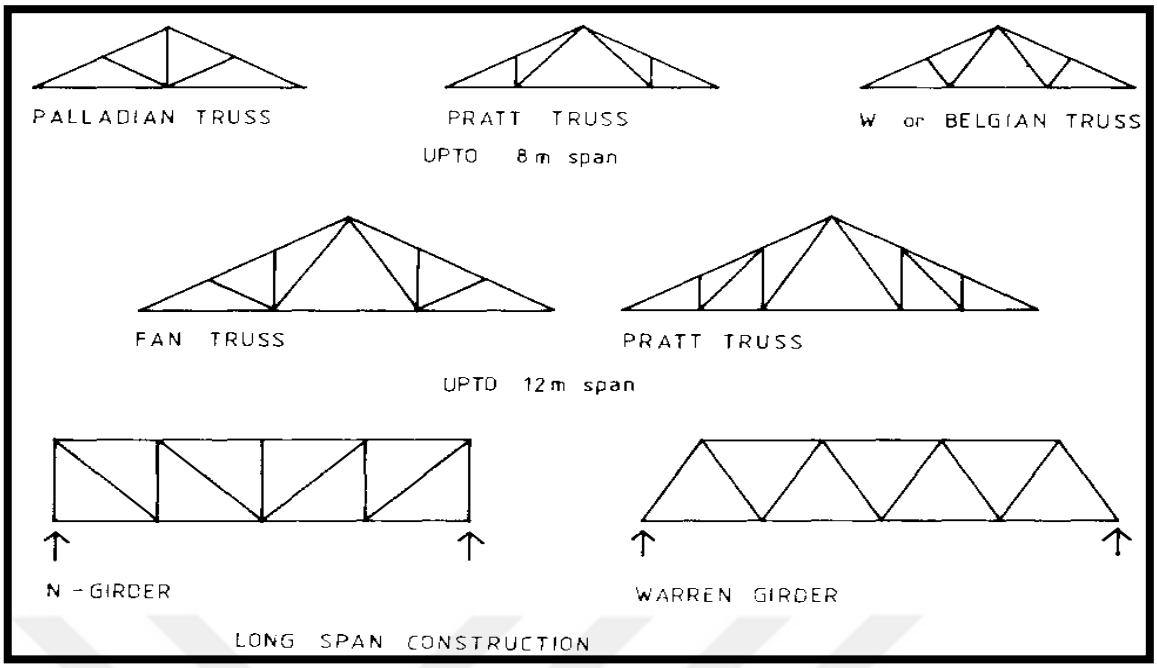
2.2 GAMS'nin Örneklerle İncelenmesi

2.2.1 Kafes-Makas Kiriş Sistemler

Statik dengeyi koruyarak üzerine etkiyen sabit veya hareketli yüklerin eksenel çekme ve basınç kuvvetleri ile kavrayan çubuk elemanların bir düzen içinde bir araya geldikleri taşıyıcı sistemlerdir.

Makas sistemler, çubuk ve düğüm noktalarının üçgen bir form düzeni içerisinde bir araya gelmelerinden oluşmaktadır. İki düğüm noktası arasında kullanılan çubuklar doğru eksenli ve sabit en kesitlidir. Düğüm noktalarında, çubuk eleman en kesitlerinin ağırlık merkezlerinden geçen doğrular bir noktada kesişmektedir. Makas sistemlerde çubuk ve düğüm noktası birleşimlerinde perçin, bulon ve kaynak kullanılmaktadır.

Makas sistemlerde, yükler yalnızca düğüm noktalarına etki etmektedir. Bu noktaların üzerine etki eden yükler, üçgen oluşturulan çubuklar tarafından taşıyıcı sistemin tamamına dağıtılmaktadır. Bu üçgenler, basınç veya çekme kuvvetlerinin etkisi altındadır. Makas sistemlerde oluşan kuvvetler moment ve kesme kuvveti şeklinde olmamaktadır. Bazı eğme kuvvetleri sistemin Üzerine dağılan yüklerin, sistemin birleşme noktalarında oluşturduğu sürtünmeden ortaya çıkmaktadır. Fakat bu kuvvetler eksenel kuvvetlerin yanında küçük bir büyüklüğe sahip olup genellikle analitik hesaplamalarda göz ardı edilmektedir.



Şekil 2.1 Kafes kiriş tiplerinden bir kaç kesit örneği [41]

Bir makas sistemin en küçük geometrik öğesi üçgendir. Üçgen strüktürel parçalar, üzerine etki eden kuvvetlere karşı çok dayanıklıdır. Ayrıca bütün çokgenler üçgen gibi stabil bir forma sahip değildir.

Düzlem makaslar, çeşitli iç çubuk bölünmeleri ve dış şekilleri arasında yapılacak olan kombinasyonlarla farklı geometrilerde üretilebilirler. Düzlem makaslar üçgen, dikdörtgen, eğrisel veya merceksi (makasın üst ve alt çubuklan eğri) olarak biçimlendirilmektedir. Şekillendirilmiş bu makaslar, küçük üçgen parçalara bölünmektedir. Bütün bu küçük üçgen parçalar birleşim noktalarından dolayı kesintili bir yapıya sahiptir.



Şekil 2.2 Bir kafes kiriş örneği (parabolik-paralel başlıklı) [42]

Makaslar, eğimli çatılar için çok uygun bir strüktürel sistemdir. Bu sistem kullanılarak çatılarda dolaylı aydınlatma yapılabilmektedir. Ayrıca makas sistem çok büyük açıldıklarda kullanılabilmektedir, örnek olarak Severin köprüsünde kullanılan makaslarla 300 m. açıklık geçilmiştir. Fabrikalar ve spor alanları düz makasın en çok kullanıldığı yerlerden bazılarıdır. Bununla beraber makasları yangına karşı yalıtmak ekonomik açıdan çok masraflıdır. Bu nedenle yangın yalıtımı gereken binalarda makas sistem kullanılmamaktadır.

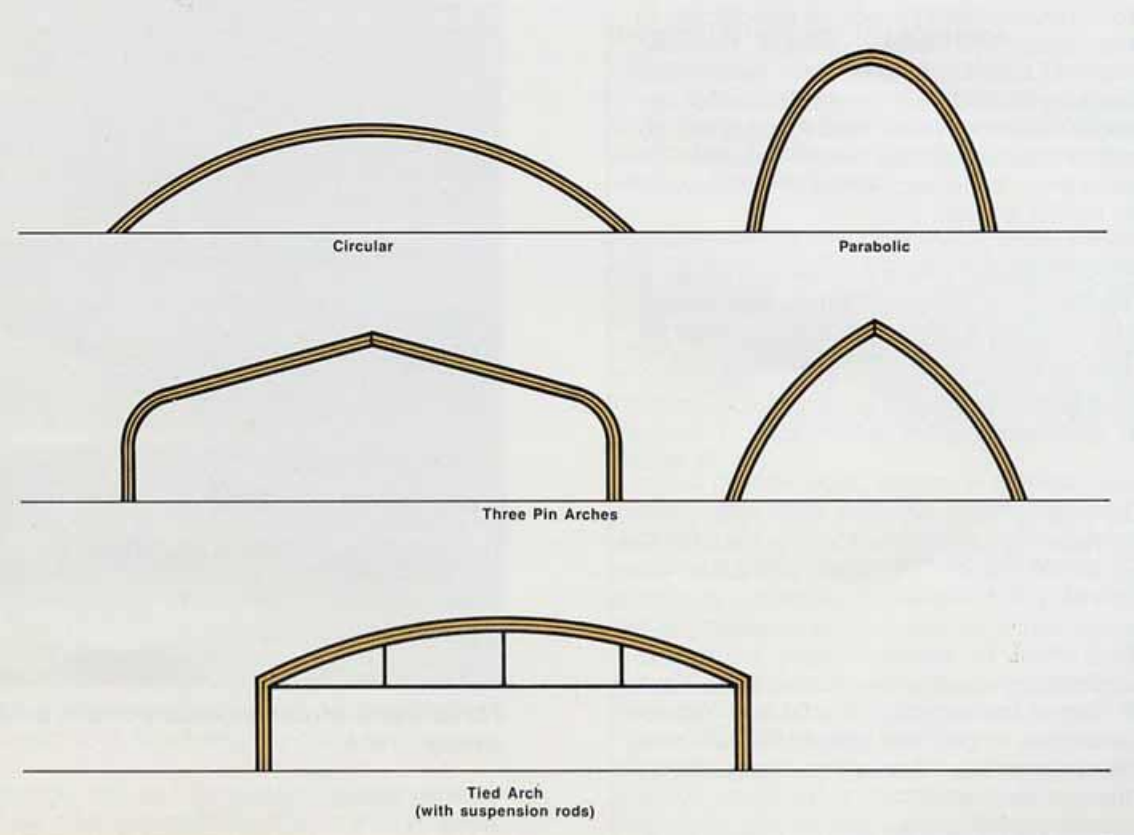
Düzlem makaslar iki boyutlu, uzay kafes makaslar üç boyutlu şekle sahiptirler. Düzlem makas ve uzay kafes makas sistemler tek bir yönde açıklık geçebilmektedir. Makas sistemlerden ayrı bir sistem olan uzay kafes çatı sistemlerde yük her iki yönde taşınmaktadır [1].

2.2.2 Kemer Sistemler

Kemerler, enkesit ölçüleri uzunluklarına oranla küçük olan eğri eksenli taşıyıcı sistemlerdir. Kemerin eğri eksenli kirişten farkı, hiperstatik (çok bağlı) olarak mesnetlenmesidir. İzostatik bir sistem olan eğri eksenli kirişte üç, hiperstatik olan kemerde ise dört tane mesnet tepkisi oluşur.

Kemerin enkesit ölçüleri, kullanılan malzemeye bağlıdır, kargir malzeme, çelik veya ahşaba oranla daha büyük enkesitler gerektirir. Tarihsel gelişimde, taş veya tuğladan

yapılmış yığma kemerler pencere ve kapı açıklıklarını aşmak için kullanıldı. Kemerler birbiri ardı sıra bitişik olarak inşa edildiklerinde Tonoz çatı adını almış ve hacimlerin yüzeysel örtüsü olarak kullanılmıştı. Geleneksel yapılarda kemerler kargir olup üzerlerindeki duvar ağırlıkları ile yüklü idi. Günümüzde ahşap, çelik, betonarme v.b. malzemeler kemerlerde kullanılarak büyük salonlar ve köprüler gibi daha büyük açıklıklar ekonomik olarak geçilebilmektedir. Salonlarda, tutkallı ahşap kemerler yaklaşık 100 m ye kadar ve betonarme kemerler de yaklaşık 50 m ye kadara açıklıklarda kullanılmıştır. Kemer köprüler ise 400 m ye kadar açıklıklarda kullanılmıştır. Kemer köprü tarafından taşınan yükler çoğu kez yatay bir yüzeye etkir. Bu yüzey kemerin üstünde ise düşey doğrultuda yerleştirilmiş basınç çubukları ile kemere bağlanır.



Şekil 2.3 Kemer kiriş sistem kesit şeması [44]

Kemer daha çok basınç gerilmelerinin egemen olduğu bir sistemdir. Kemerde sadece basınç gerilmeleri oluşması için, basınç kemeri (ya da füniküler kemer) adı verilen kemer biçimi yüklemeye uygun şekilde seçilir. Buna göre her bir yük durumu için belirli bir basınç çizgisi vardır. Örneğin, düzgün yayılı yük için 2.° parabol olan basınç çizgisine ip

eğrisi adı da verilir. Tekil yükler için yük sayısına bağlı olarak üçgen, yamuk ya da çokgen olan basınç çizgilerine ise ip çokgeni denir.

Kemer ekseninin basınç çizgisinden sapması durumunda ise kemerde ek eğilme gerilmeleri ortaya çıkar. Çünkü bir kemer sadece sabit yükler göz önüne alınarak basınç kemeri olarak seçilse bile, hareketli yükler etkidiğinde kemerde yine eğilme momenti meydana gelebilir. Bu nedenle uygulamadaki kemerler, basınca ve eğilmeye çalışan taşıyıcı sistemlerdir.

Rijit kemerler binalarda sıklıkla kullanılırlar. Kemer sistemlerinin üç mafsallı, iki mafsallı ve ankastre kemer türleri vardır. Bunlardan üç mafsallı kemerler izostatik olduğu için, mesnet oturmalarının oluşabileceği yapı zeminlerinde de kullanılabilirler. Başka bir deyişle üç mafsallı kemerler, sıcaklık değişiminden ve temel oturmalarından etkilenmezler.



Şekil 2.4 Kemer kiriş sistem örneği – Atina Olimpiyat Stadı Girişi [45]

Kemerlerdeki H yatay mesnet tepkisi “kemer itkisi” adını alır. Kemer itkisi, kemer açıklığının karesi ve q yükü ile orantılı olarak artar, kemer yüksekliği arttıkça azalır. Bu nedenle elden geldiğince, kemerin öz ağırlığı az ve yüksekliği de büyük seçilerek bu kemer etkisi küçültülebilir, örneğin yüksekliği fazla olan sivri kemerdeki etkiler, yarım daire kemere oranla daha küçük olur, kemerin oturduğu kolon veya duvarların kemer

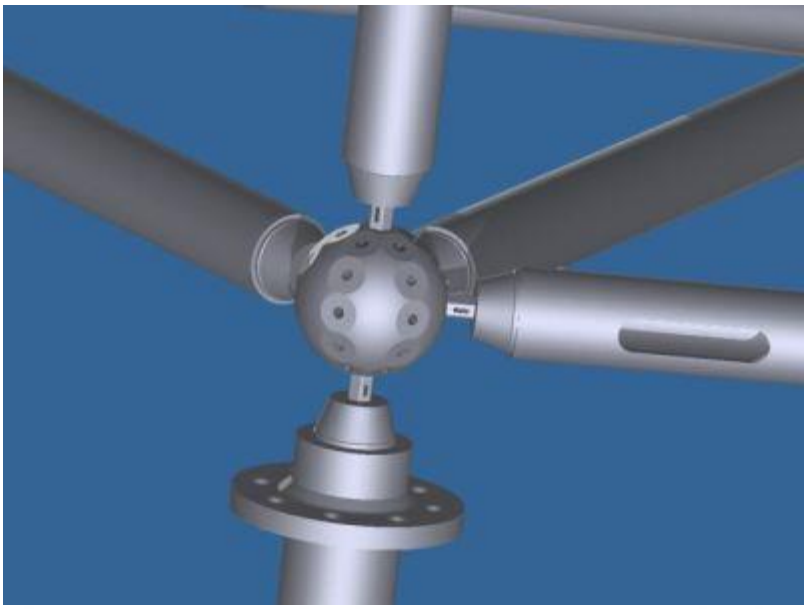
etkilerini taşıyamadıkları durumlarda, kemerin mesnetlerinin arasına bir gergi konarak kemer etkisi bu gergi çubuğu tarafından karşılanır. Bu gergi çubuğu, S çekme kuvvetini taşıyacağı için çekme dayanımı yüksek bir malzemeden, örneğin çelik ya da betonarmeden yapılır. Böylelikle bir kafes sisteme benzer şekilde basınç ve çekme elemanlarının birlikte kullanıldığı bir kemer-gergi birleşimi oluşturulmuş olur [2].

2.2.3 Uzay Kafes Sistemler

Uzay kafes strüktürler, çekme kuvvetine çalışan doğrusal çubuk biçimindeki öğelerle, yükleri çok yönlü bir yayılım sonucu zemine aktaran, boşluğun organize edilmesi ilkesine göre üretilen çağdaş strüktür sistemleridir.

Uzay kafes strüktür, büyük açıklıkların yatay düzlem çatı ile kaplanmasında sıklıkla kullanılmaktadır. Ayrıca uzay kafesler büyük açıklıklı mekanların üzerinde eğimli olarakta yerleştirilebilmektedir. Teknolojinin ilerlemesiyle birlikte bu sistem 100m'yi geçen açıklıkları örtebilmektedir. Sistemin en geniş kullanım alanları, spor salonları, uçak hangarları ve endüstri yapılarıdır. Ayrıca fuarlarda dev boyutlardaki pavyonların taşıyıcı sistemleri de uzay kafeslerden yapılmaktadır.

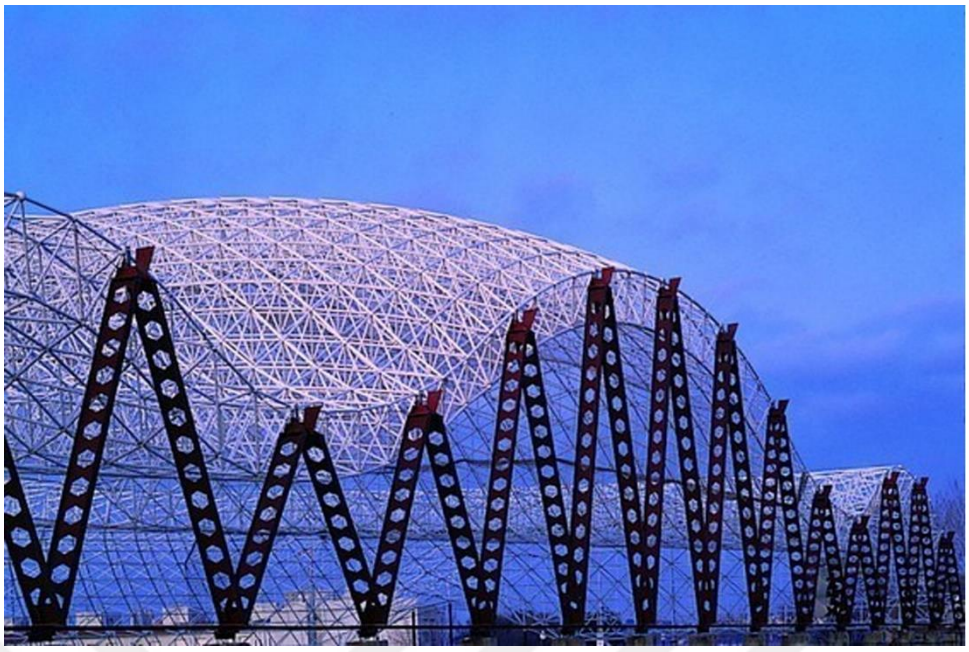
Uzay kafes sistemlerin yapım tekniklerinin gelişmesiyle birlikte, bu taşıyıcı sistemlerin montajı kolaylaşmıştır. Sistem yerde kurulup, vinçlerle yerine monte edilebilmektedir. Hatta montaj işlemi çatı örtüsü hareket ettirilmeden yerinde yapılabilmektedir. Sistem prefabrike parçalardan oluşmaktadır. Bu nedenle gerektiğinde sökülüp başka yerlere taşınabilmektedir.



Şekil 2.5 Uzay kafes sistem örnek detay[46]

Uzay kafeslerin montaj kolaylığı, kalıp ve iskele ihtiyacının olmaması inşaatın maliyetini azaltmaktadır. Ayrıca uzay kafes sistemler, diğer taşıyıcı sistemlere nazaran çok hafif strüktürel sistemlerdir. Bu nedenle taşıyıcı sistem öğeleri ve temeller daha az bir maliyet ile inşa edilebilmektedir. Buna karşın uzay kafes sistemlerin düğüm noktaları bileşenlerinin üretimi ve tasarım hesaplarının uzun sürmesi ile birlikte yüksek iş gücü gerektirmesi yapının maliyetini arttırıcı faktörlerdir.

Uzay kafes strüktürler, değişik formlarda inşa edilebilmektedir. Düzlem yüzeyler, tonoz ve kubbe biçimleri uzay kafes ile yapılabilmektedir.



Şekil 2.6 Uzay kafes sistemle oluşturulmuş örnek form [46]

Uzay kafes sistemlerin ekonomik derinliği %5 kadardır. Açıklık geçilen sistemde konsol kullanılmışsa, uzay kafesin derinliği bu konsolun %11'i kadardır. Uzay sistemlerde ekonomik modül boyutu, geçilen açıklığın %7 veya %14 'ü kadardır.

Bu sistemler ile makas aşık sistemler karşılaştırıldığında, uzay kafeslerin iki veya daha fazla doğrultuda yük taşıdıkları görülmektedir. Bunun tam tersi olarak iki boyutlu makas sistemler tek bir yönde açıklık geçebilmektedir.

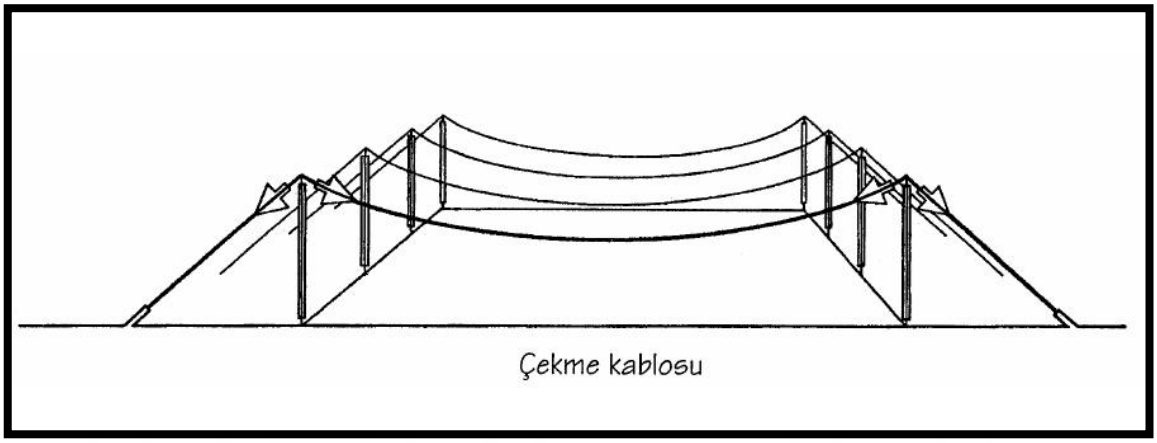
Uzay kafesler, etkili ve güvenilir strüktürel sistemlerdir. Taşıyıcı sistemi oluşturan her bir elemanın mukavemeti ile doğru orantılı olarak üzerlerine kuvvetler etki etmektedir. Uzay kafesin güvenliği, sistemden birkaç strüktürel elemanın eksilmesiyle değişmez. Sadece bazı kuvvetlerin yönü değişir. Diğer elemanlar, dirençleri ile doğru orantılı olarak eksilen strüktürel elemanlara etki eden kuvvetleri paylaşmaktadır. Üzerine direnç gösterebileceği kuvvetten daha büyük bir kuvvet etki ederse, bu sistemlerin dayanıklılığı ve emniyeti belli bir dereceye kadar devam etmektedir. Ancak üzerine etki eden aşırı yüklemeler sonucunda bazıları çökmüştür.

Hartfort Civic Center'ın üzerini kaplayan (90 m. x 110 m.)'lik uzay kafes strüktürel sistem, çatının üzerinde aşırı kar toplanması yüzünden yıkılmıştır. Daha sonradan yapılan analizlerde, 6.4m'lik derinliğe sahip olan uzay kafes çatının iyi bir şekilde çapraz

bağlanmamış olması, uzay kafes elemanların zaman içerisinde burkulmasına ve taşıyıcı sistemin yavaş yavaş çökmesine yol açmıştır [1].

2.2.4 Asma Germe Sistemler

Yapının ana yüklerinin, ankraj noktaları arasına gerilmis çekme elemanları tarafından tasındığı tasıyıcı sistemler kablo sistemler olarak tanımlanır. Yani çok sayıda kablonun çesitli geometrilerle bir araya getirilip, düğüm noktalarıyla birlestirilerek tek ve çift eğrilikli ayrık yüzeyler olusturulduğu ve sadece çekmeye çalısan sistemlere kablo sistemler denir.



Şekil 2.7 Kablo sistem

Doğal liflerden yapılmıs halatlarla gerçeklestirilen asma köprüler, kablo sistemlerin ilk örnekleri sayılabilir.1976 yılında James Finlay demir zincir halkalarıyla tasınan küçük asma köprüler yapmıstı. Bugünkü kullanıma yakın kablolu yapıların gelisimi 19.yy da endüstri devrimi ile ortaya çıkan çelik halat kabloların üretilmesiyle olusmaya baslamıstır. Çelik kabloların tasıyıcılık özellikleri kazanmaları öncelikle asma köprülerde gerçeklesmistir. 19. yüzyılın sonlarında kablonun ana tasıyıcı olduğu asma bir çatı imal edildiği bilinmektedir. Bu konuda ilk önemli uygulama Nowicki' nin 1952 yılındaki Raleigh Arena Binası kabul edilmektedir. Bundan sonrada hızla uygulamalar artması özellikle genis mekanların örtülmesinde çekmeye çalısan sistemlerin çağı olarak isim kazanmasına neden olmustur.

Kablo sistemlerin hafifliği, kabloların yüksek dayanımı, çekmeye çalısan elemanların burkulma sorununun olmaması ve bu nedenle malzemenin el verdiği kadar ince yapılabilmesi bu sistemin avantajlarıdır. Ayrıca eğri biçimlerin yarattığı strüktürel

üstünlüklere ek olarak kabloların yalnızca çekme etkisinde olmaları asma sistemlerin büyük açılıkların kolon gerektirmeden geçilmesinde en etkili sistemlerden biri durumuna getirmektedir.

Asma-germe sistemler geleneksel çelik sistemlerden 10 kez, dolu gövdeli iskelet sistemden ise 100 – 150 kez daha hafif oluslarıyla, büyük açıklıklı yapılarda önemli ekonomik çözümler getirmektedir. Çekmeye çalısan çelik kabloların ve çelik ipliği ile güçlendirilmis ince membran (tekstil ürünü gerilmis tabaka) örtülerin kesit ölçülerinin çok az olması nedeniyle (1 – 3 mm), bu strüktürler hafiflik açısından büyük bir etkinlik ortaya koymustur. Bu sistemlerde kalıp ve iskele yok denecek kadar az kullanılmaktadır. Bu strüktürlerin 40 m' den az açıklıklarda uygulanmaları ekonomik olmamalarına karsın, açıklıklar arttıkça ekonomikliği de doğru orantılı olarak artmaktadır.



Şekil 2.8 Asma germe sistem örneği – Millenium Dome

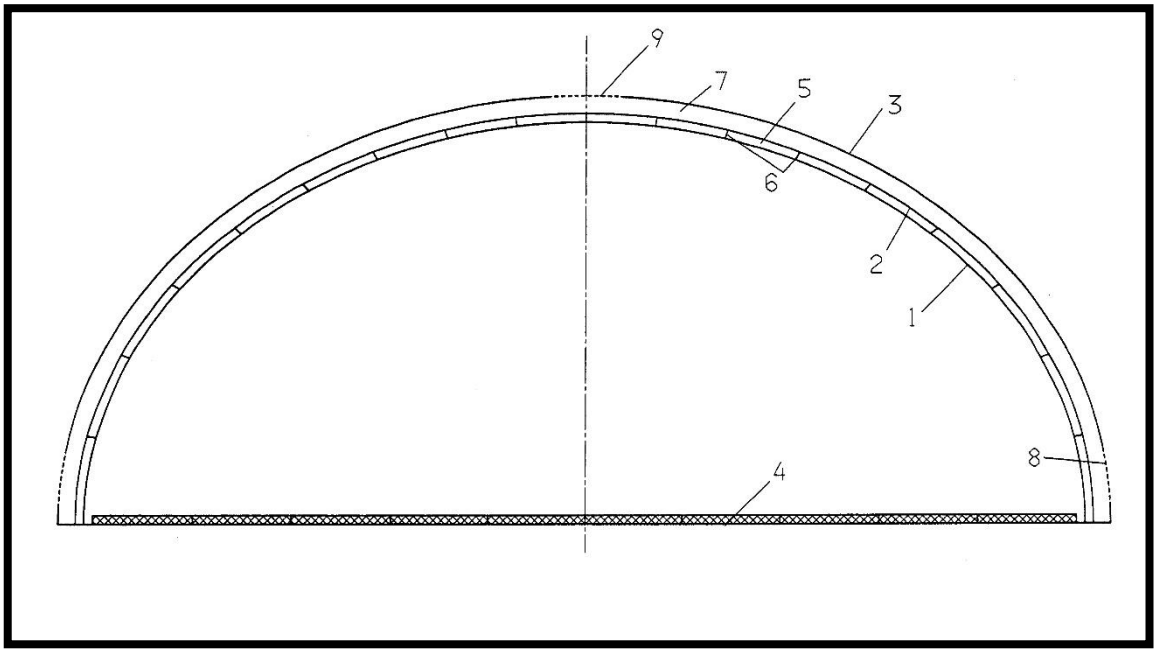
Kablolar, bir tasıyıcı sistemin elemanları olabilecekleri gibi, membran sistemlerde yüzeyleri germek veya iskelet sistem gibi baska bir tasıyıcı sistemin yardımcı

elemanı olarak büyük konsolları uçlarında bir çekirdeğe asmak gibi baska amaçlarla da kullanılmaktadır [47].

2.2.5 Pnömatik Sistemler

Kapalı membranlar olarak da tanımlanan grupta membran hem örtme hem de tasıma görevlerini birlestiren bir yüzey olup tamamen kapalı bir hacim olusturur. Kapalı bir hacim içerisinde yaratılan basınç farkı ile membran örtüye öngerilme verilerek sistemin tasıyıcılığı sağlanır. Pnömatik sistemlerin mimarlıkta taşıyıcı örtü olarak kullanılmaya başlanması çok yenidir. Başka işlevler için güzel örnekler arasında geçen aırlarda bilinen balonlar ve 20. Asır başlarında kullanılan zeplinler sayılabilir. Canlı organizmanın temel taşı olan hücrelerde de pnömatik ilkeler, yani içteki basınçla bir cidarın şekillenmesi mevcuttur. Bütün canlılar, milyonlarca hücrenin bir araya gelmesiyleoluştuğuna göre; tüm yaşayan strüktürler, biz zamanlar pnömatik ilkelere göre gelişmiş çok sayıda strüktür elemanlarının kümelenmesiyle ortaya çıkan karmaşık sistemlerdir.

Pnömatik bir sistemde yüzey gerilmesi, iç hacimle dış hacim arasında basınç fazlası veya basınç eksikliği yaratılarak sağlanır. Basınç fazlası için kapalı bir hacmin içine, akışkan bir madde veya gaz doldurulur.



Şekil 2.9 Çift cidarlı pnömatik sistem kesit şeması [48]

Kapalı bir hacim içinden hava boşaltılacak olursa, atmosferin basıncı ile yüzey içeri doğru itilir. Mimarlıktaki uygulamalarda genellikle basınç fazlası yaratmak yaygın ve olağandır. Bunu sağlamak için başvurulan en yaygın yöntem, bir veya iki cidar içine hava pompalamaktır. Diğer taraftan su, kum vb. gibi akışkanların veya yarı akışkan maddelerin kapalı bir hacme (örneğin çuval veya torbaya) sıkıştırılması da o hacimde yüzey gerginliği yaratır ve daha önce yük taşıyamayacak bir esnek malzemeli yüzey, yüklere dayanabilecek bir konuma gelir.

Pnömatik sistemler söyle sıralanabilir;

Basınç düzeyi: Alçak veya yüksek basınçlı sistemler.

Tabaka sayısı: Tek veya iki cidarlı sistemler.

Basınç farkının sisirme veya emme yoluyla sağlanması.

Geometrik biçim: Tek boyutlu sistemler veya hacimsel sistemler.



Şekil 2.10 Çift cidarlı-kanallı pnömatik sistem örneği – BC Place Stadium [49]

Pnömatik sistemlerde biçimin seçimi önemlidir. Biçim mimari estetiğin yanı sıra seçilen geometrinin membran sistemler tarafından yapılabilirliğini, yüzeyde olusacak gerilmenin büyüklüğünü yani statik davranısını etkilemektedir [50].

2.3 Parametrik Tasarım Kavramı

Sözlük anlamı olarak:

Parametre: isim, matematik (parame'tre) Fransızca paramètre

1. Cebirde bir denklemin kat sayılarına giren değişken nicelik.
2. Geometride, bir koninin odağından çıkan dikeyin konikle kesiştiği noktaya kadar olan parçanın uzunluğu.

Parametrik: Parametrelere sahip olan

En yalın anlatımıyla parametrelere bağımlı tasarımdır. Parametre bir bağımlılık belirtmektedir. Tasarımcıyı, dijital medya ile birlikte ortaya çıkan yeni tasarım dilinin mantığı ve tekniği ile tanıştırır. Sayısal ortamın hesaplamalı (computational) düşünme ortamı olarak sunduğu farklılaşan tasarım yaklaşımlarını ve mimari açılımlarını, bilgisayar ortamı ile içinde yaşadığımız fiziksel gerçeklik arasındaki ilişkiyi vurgular.

Parametrik tasarımda, tasarım parametreler aracılığıyla tanımlanmıştır. Parametrelere verilen farklı değerlerle, farklı objeler ya da farklı düzenlenişler ortaya çıkabilir. Parametreler arasındaki ilişkiyi şekillendirmek için denklemler, eşitlikler kullanılmaktadır. Parametre ilişkileriyle tanımlanmış geometriler, "ilişkilendirilmiş geometriler" olarak adlandırılır[4].

Parametreler değiştirildikçe farklı tasarım alternatifleri oluşmaya başlar, benzer objelerin kesin sayısına ulaşılır. Parametrik tasarım ve insan beyninin kendi başına üretebileceği alternatif sayısının üzerinde alternatif üretilebilir. Yapılan küçük bir değişiklik, hazırlanan algoritma sayesinde projenin bütününe yansıtıldığı için, değişiklik nedeniyle oluşabilecek zaman kaybının önüne geçilebilir.

Bilgisayar ortamında kurgulanan parametrik tasarım sistemi, parametrelere girilen farklı değerlerle, tasarım aşamasında biçim üretimi için ya da fiziksel mekânda ışık-ses-biçim değişimleri için kullanılabilmektedir [5].

Geometri her zaman mimari söylemde önemli bir rol oynamıştır. Mimari tasarımlar, parametrik modelleme ve dijital hesaplamaların araç olarak kullanıldığı dinamik & geometrik ilişkilere dönüşmüştür [6].

Mimarlar başlangıçta, bir yapının seklini parametrelerle tanımlanan geometrik ilişkileriyle kurmaktan uzaktaydılar. Parametrik tasarım yöntemleriyle, tasarımlar değişkenlere, ifadelere ve koşullara bağlı kalarak kurulan parametrik denklemlerle tanımlanmaya başlanmıştır. Yine parametrik denklemlerin belirlenen değişkenleri,

ifadeleri, bağıl söylemleri ve komut dizinleri (scripting) değiştirilerek ayrıntılı tasarım örnekleri türetilebilmektedir

Parametrik tasarım yöntemi, CAD/CAM sistemlerinde kısıtlılık (constraint) kavramı ile uygulanmaktadır. Kısıtlılık, bir nesne veya nesneler grubunun davranışını sınırlandıran bağıntılardır. Çizgilerin paralel veya dik olmaları, bir çizginin bir eğriye teğet olması, dairelerin eş merkezli olmaları, bir boyutun başka bir boyuttan küçük olması veya bir boyutun bir başka boyutun belirli bir katı olma zorunluluğu kısıtlılığa örnek olarak gösterilebilir [7].

Güncel mimarlık tartışmalarındaki parametrik tasarım kavramı, bilgisayar ortamı ile fiziksel gerçeklik arasındaki ilişkiyi anlatır. Parametrik tasarım bilgi ve değişkenlerden oluşan bir hesap çizelgesini ve şartları, senaryoları, etki ve tepkilerden oluşan eylemleri içeren bir tasarım setinin tümünü oluşturuyor. Bu model potansiyelleri bir araya getiren bir arayüz oluştururken, tasarım da uzlaşan kuralları belirler. Örneklemeleri çoğunlukla üçboyutlu modellemelerden ve maketlerden oluşur ve genelde bilgisayar teknolojileri kullanılmaktadır. Pratik olarak parametrik tasarım, malzemeden kullanıcıya tüm iç ve dış etkenlere cevap veren bir evreni içerir. Kullanıcılardan ve/veya çevreden geri beslenerek işler. Bu nedenle, örneğin mimarlıkta yapının tasarımına cevap veren tek bir tip veya program yoktur, süregelen bir değişim vardır. Yapılan örneklerde bu süreklilik genel olarak statik/durağan bir hali hayat bulmuştur. Parametrik tasarım geçicilik içerir ve mimarlığı zamanla değişen esnek bir organizasyon olarak görmektedir.

Parametrik tasarımda; özellikler, bağımlılıklar, ilişkiler, kontrol noktaları, süreç tasarımın bir parçası olmadan önce sürecin stratejileri, etkileşimin sonucunda ortaya çıkan ilişkili ve beklenmedik sonuçlar bulunur. Bu sayede çok fazla alternatifli ve zaman içinde değişkenlik gösteren ve girdileri ile değişebilen bir mekan organizasyonu elde edilmektedir.

Parametrik tasarım, forma ilişkin hesaplanabilen bir formüller bütünü ve metinden oluşur. Program ya da lokasyon bağlamlarının değişken hallerine adapte olabilen bir formdur bahis konusu olan, dolayısı ile strüktürün de adapte olan, benzer prensiplere ve değişen şartlara adapte olmuş, olabilen strüktürler oluşturulabilir. Aslında parametrik strüktür tasarımının, konvansiyonel mimari pratiğindeki hiyerarşileri değiştireceğini

söylemek mümkün. Parametrik strüktürün farklı noktalardaki farklı güç taleplerine göre biçim alabilmesi ve daha hafif ya da daha yerinde olması söz konusu. Parametrik strüktür yüzey, kabuk, form boyunca benzer eleman ve prensiplerle birbirine bağıntılı ama değişken boyut ve şekillerle açıklık geçmek, mekan kurmak için oluşturulan yapıyı tarif ediyor.

Geometri her zaman mimari söylemde önemli bir rol oynamıştır. Mimari tasarımlar parametrik modelleme ile dijital hesaplamaların araç olarak kullanıldığı dinamik geometrik ilişkilere dönüşmüştür. Mimarlar başlangıçta, bir yapının seklini parametrelerle tanımlanan geometrik ilişkileriyle kurmaktan uzaktaydılar. Parametrik tasarım yöntemleriyle, tasarımlar değişkenlere, ifadelere ve koşullara bağlı kalarak kurulan parametrik denklemlerle tanımlanmaya başlanmıştır. Yine parametrik denklemlerin belirlenen değişkenleri, ifadeleri, bağıl söylemleri ve komut dizinleri (scripting) değiştirilerek ayrıntılı tasarım örnekleri türetilebilmektedir.

Parametrik tasarım yöntemi, CAD/CAM sistemlerinde kısıtlılık (constraint) kavramı ile uygulanmaktadır. Kısıtlılık, bir nesne veya nesneler grubunun davranışını sınırlandıran bağıntılardır. Çizgilerin paralel veya dik olmaları, bir çizginin bir eğriye teğet olması, dairelerin eş merkezli olmaları, bir boyutun başka bir boyuttan küçük olması veya bir boyutun bir başka boyutun belirli bir katı olma zorunluluğu kısıtlılığa örnek olarak gösterilebilir [7].

2.3.1 Parametrik Modelleme

Parametrik modelleme, parametrelere dayalı olarak kurulan ve sonuç biçimin parametrelerin kontrolü altında sürekli değişebildiği modelleme yöntemleridir. Parametrik modellemede tasarımı geometrik olarak sınırlayan kurallar belirlemektedir.

Parametrik modelleme yazılımlarının öncülleri uçak, otomobil, gemi-inşaat mühendisliği ve üretim-tasarım endüstrisi için geliştirilmiştir. Mimaride bu modelleme yöntemiyle karmaşık geometriler düzenli bir rasyonalizasyon (mantığa vurma) sürecine sahip olabilmektedir. Parametrik tanımlama sonrasında yapının genel kurgusu üzerinde hızla değişiklikler yapılabilmektedir. Parametrik model, üretim ve konstrüksiyon ile ilgili ilişkisel ve geometrik bilgileri de kapsayabilmektedir. Dolayısıyla, üretim ile ilgili

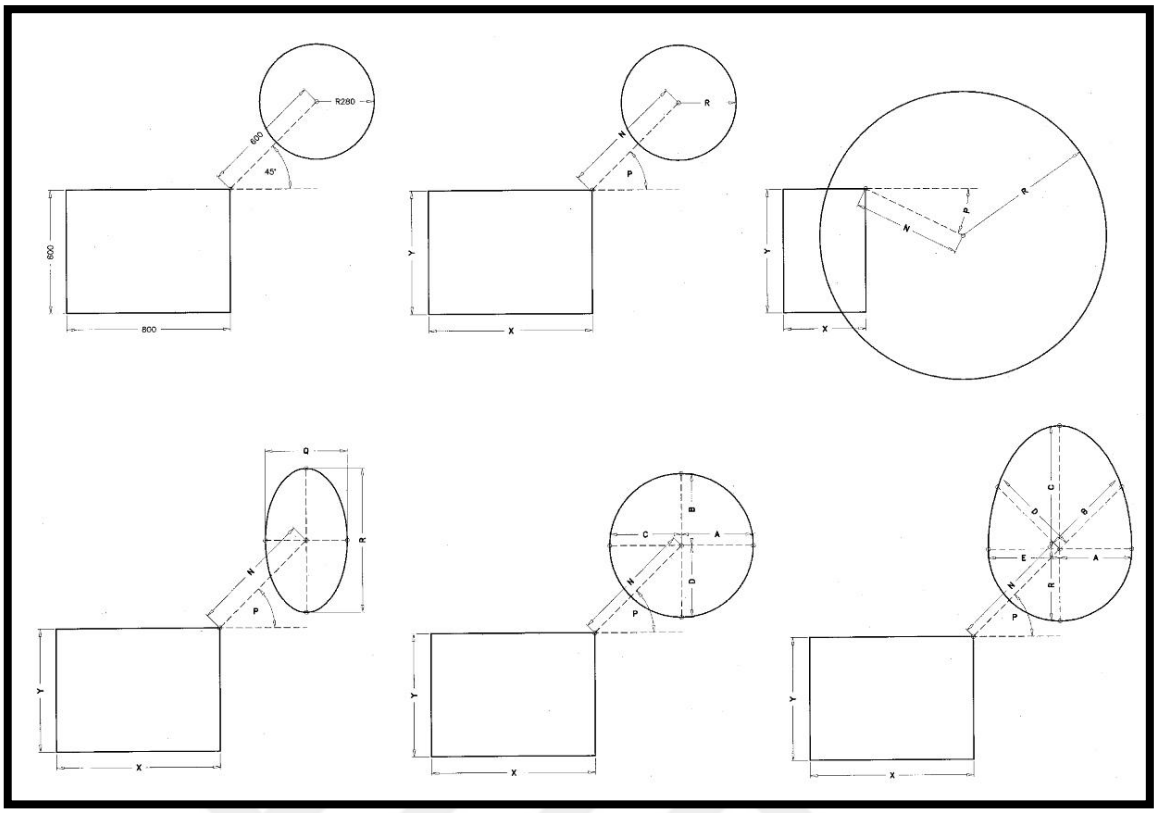
bilgilerde parametrik model üzerinden gerektiğinde değişen verilerle yeniden hesaplanabilmektedir [6].

Parametrik modellemelerde hesaplamalar, geleneksel tasarım ve yapım süreçlerinde olduğu gibi statik bir geometri kurgusuna sahip değildir. Tasarımcı parametrik modellemede, tanımlanabilen ve değiştirilebilen geometrik bir sistemi geliştirme durumundadır. Parametrik modellemede sistemin ölçülebilir ve geometrik özelliklerinin kurallarının tasarlanmış olması gerekir.

Parametrik modelleme topolojik geometriyi temel alır. Çünkü Öklid geometrisi, uzunluk, alan ve hacim gibi kavramalara dayalı, metrik bir uzayda işlerken, topolojik geometri uzayında mesafe, yakınlık ile tanımlanamayan ve sabit kalmayan, dinamik bir uzunluk cinsi ile ifade edilmektedir. Topolojik uzayda objeler, tanımlanan noktaların karakteristik özellikleri değiştirilmeden esnetilebilir ya da objelerin ölçeği değiştirilebilir.

Parametrik tasarımda bir modelin temeli noktalardan oluşmaktadır. Öklid geometrisinde noktalar arasındaki ilişki, sabit uzunluklar ve ölçülerle tanımlanırken, parametrik modelleme bir noktanın diğer noktalardan ne kadar uzakta olduğu tanımlanarak kurulur.

Parametrik modelde geometrik ilişkiler tanımlanmaktadır. Eski CAD yazılımlarının "Tanımlanmış (explicit)" koordinat sistemlerine dayalı geometrisindeki model ile parametrik modelleme yazılımlarının "ilişkisel" geometrisi arasındaki fark, iki çemberin merkez noktalarından bağlanan bir doğrunun dijital çiziminin kavranmasıyla anlaşılabilir. Koordinat sistemlerine dayalı "Tanımlanmış (explicit)" geometride, her iki çemberin pozisyonunda değişiklik olursa, bağlanan çizginin silinmesi ve yeniden çizilmesi gerekmektedir. Parametrik modelde ise merkezler ve çizginin arasında öyle bir ilişki kurulur ki, çizgi her iki çemberin her hareketini takip etmektedir. Örneğin çizginin ve çemberin yarıçapının uzunluğu gibi olan diğer ilişkiler öngörülmüş olabilir ya da her iki çemberin kesişmesinin önlenmesi gibi şartlara bağlı ifadelerden sonra ilişki kurulabilir. Önceden tanımlı ve modeli çalıştıran bütün geometrik ilişkiler, parametrik olarak artan tutarlı ifadeler ve kısıtlar yolu ile tanımlanmaktadır.[6] Bu örnekten yola çıkarak "Tanımlanmış (explicit)" geometride Kartezyen uzayındaki transformasyonların statik(tek/sabit) bir etki, parametrik geometride ise ilişkisel parametrelere dayanan transformasyonların dinamik(birçok/çeşitli) etkiler yarattığı söylenebilir [7].



Şekil 2.11 Bir dikdörtgen ile bir çember arasındaki ilişkisel transformasyonları gösteren bir örnek [8]

Sekil 2.11'de Burry ve Murray'ın "Bilgisayar Destekli Mimari Tasarımın Kullanımı

Parametrik Varyasyonlar ve İlişkisel Geometri (Computer Aided Architectural Design Using Parametric Variation And Associative Geometry)" adlı makalesinden alınan bir dikdörtgen ile bir çember arasındaki ilişkisel değişimleri (transformasyonları) gösteren bir örnek yer almaktadır.

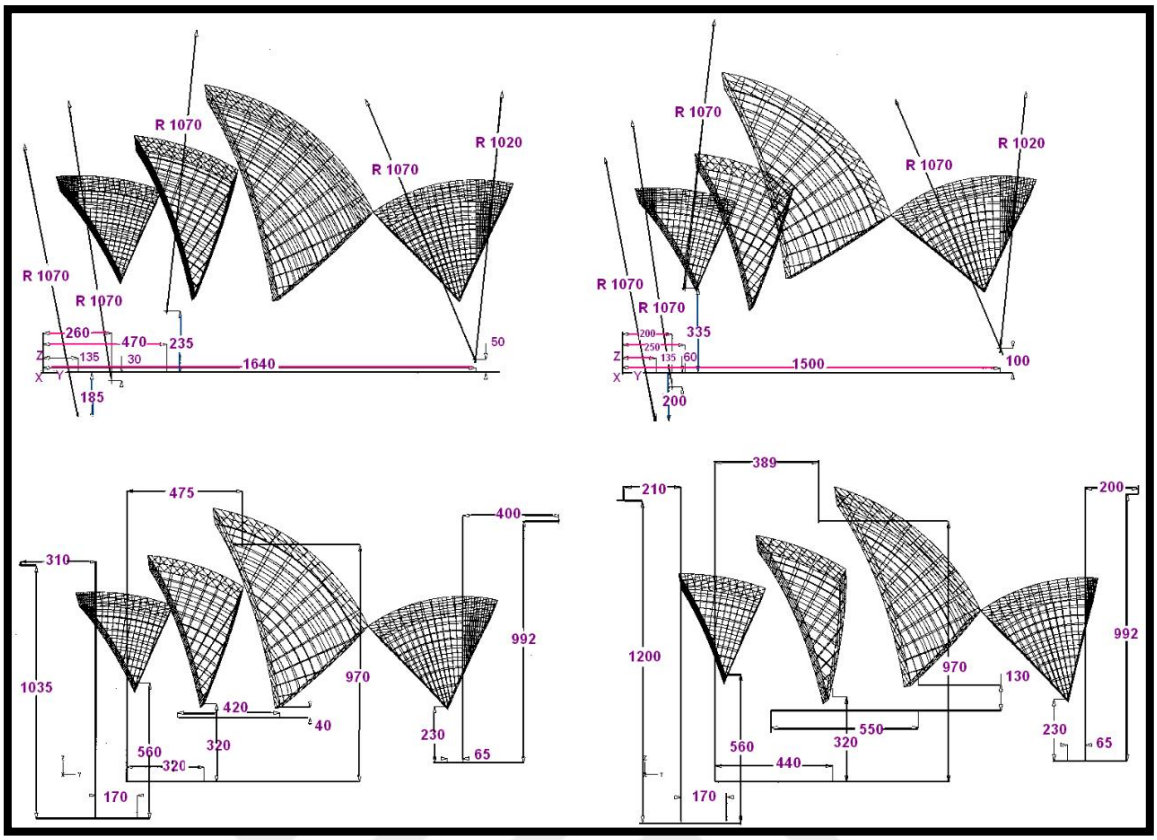
Mark Burry, parametrik geometriye dayalı tasarım sürecinde sınırların belirlenmesi ve parametrelerin sonuç ürünün biçimini istenilen düzeyde çeşitlendirebilecek nitelikte kurgulanması gerektiğini vurgulamaktadır [8].

Burry'ye göre (1997) bir yapıyı ya da mekânı parametrik olarak modelleyebilmek için;

-ilk önce sabitleri ve değişkenleri tanımlamak gerekmektedir.

-Mekanın ya da yapının belirli sayıda, birbiri ile ilişkili alt sistemlerden oluşan, düzenli bir geometriye sahip olması gerekmektedir.

-Mekan veya yapının hesaplanabilir ilişkilendirmeleri olmalıdır.



Şekil 2.12 Sydney Opera Evi'nin parametrik modellemesi [8]

Parametrik modeli oluşturmak basta biraz karmaşık olsa ve zaman alsa da, herhangi bir veri değişiminde tüm modelin değişmesi kolayca mümkün olmaktadır. Bu yenilenme dakikadan daha hızlı, saniyelik bir istir. Parametrik model, değiştirilebilir ve analiz edilebilir. Parametrik tasarım teknolojileri ile mevcut geometrilerin farklı kullanımları gündeme gelmektedir [7].

2.3.2 Mimaride Kullanılan CAD Programları

Mimarlık alanında kullanılan pek çok araç içerisinde kuşkusuz bilgisayarlar, soyut kavramlardan görsel anlatım ortamına geçişte kullanılan en yaygın araçlardır. İki boyutlu anlatımlarda kütle, çizgiler ya da yüzeyler kullanılarak oluşturulur. Üç boyutlu anlatımlarda ise kütle modelleri, iskelet modelleri yardımı ile çeşitli düşünsel tasarım önerileri, görünür ve hatta içinde dolaşılabilir duruma getirilebilir. İki boyutlu çizimlere ve üç boyutlu modellere fotoğraf, video gibi araçların görüntülerinin eklenmesiyle birlikte, tasarımın yaratılmak istendiği çevrenin, tasarımla bir arada görülmesi sağlanabilir. Sadece monitör görüntüsü olarak iki ve üç boyutlu görüntülerin yer aldığı

bilgisayarlar ile günümüzde yeni gelişmekte olan ve tamamen gerçek üç boyutlu görüntüyü lazer ışınlarının yardımı ile gerçekleştiren hologramlar, tasarı düzeyindeki çözümlerin gösterilmesini olanaklı kılan ve özellikle mimarlıkta yeni tasarım araçları arasına katılan, yeni teknolojik yardımcılardır [9].

Gelişen teknolojiyle beraber, kendini sürekli geliştirmek zorunda olan ve geliştiren mimari programlar, aynı zamanda mimari teknikleri ve mimari oluşumları da takip edip onlara da uyum sağlamak zorunda. Parametrik alt yapısına sahip, başlıca mimari programlar:

-Allplan BIM

-ArchiCAD

-Artlantis

-AutoCAD

-AutoSketch

-Cinema 4D

-Form-Z

-Google Sketch Up

-GStarCAD

-IdeCAD Mimari

-Maya

-Navisworks

-Pytha

-Revit Architecture

-Rhino

-SolidWorks

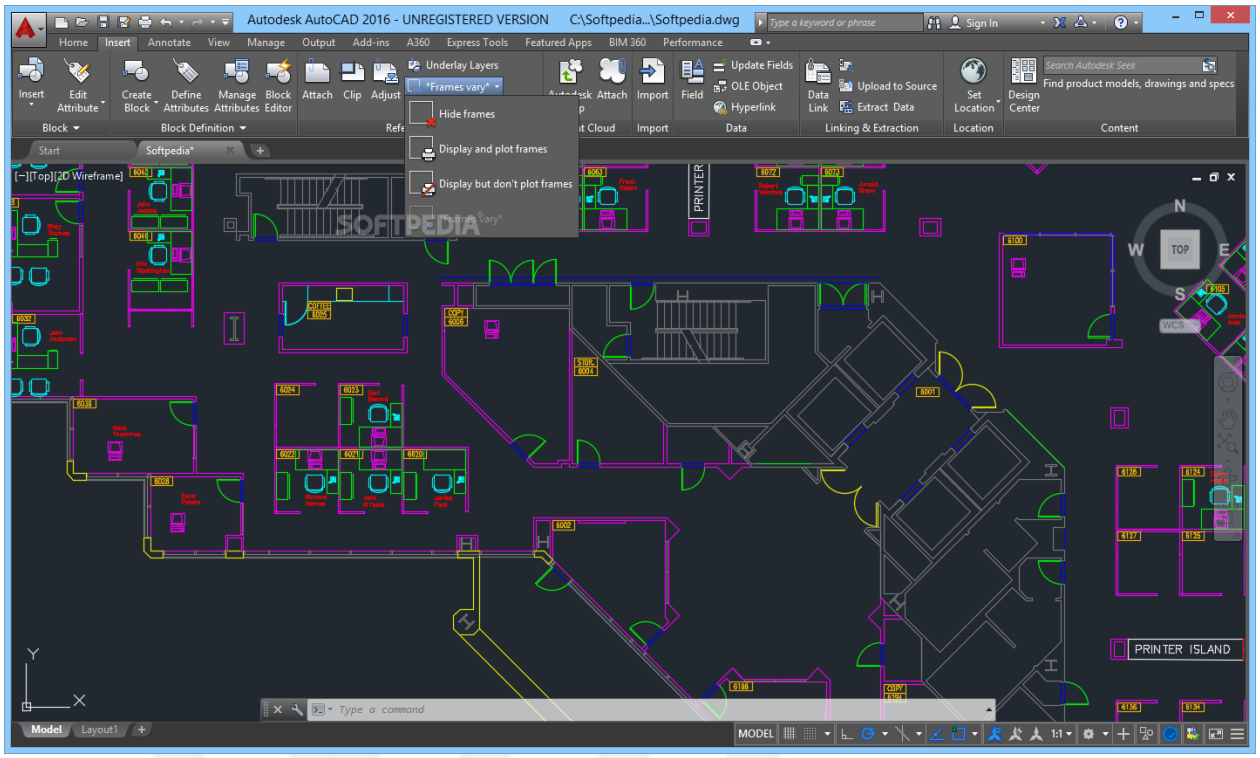
-3ds Max

2.3.2.1 AutoCAD

Autodesk, 30 yıldır tasarım süreçlerine etkinlik getirmek için çalışmaktadır. Dünyanın her yerinde milyonlarca kullanıcının tercihi olan AutoCAD aynı zamanda dünyanın önde gelen tasarım ve dokümantasyon çözümlerinden birisidir. AutoCAD yazılımı 3B serbest form tasarım araçlarından güçlü çizim ve dokümantasyon fonksiyonlarına kadar pek çok özelliğe sahiptir ve tasarımda yenilikçiliğin öncüsü ve itici gücüdür.

Tasarım dokümantasyonunu hızlı bir şekilde hazırlayabilir, tasarımları doğru olarak paylaşabilir ve fikirleri 3B ortamında daha sezgisel olarak araştırabilirsiniz. AutoCAD yazılımı, ihtiyaçlarınıza uygun hale getirilebilir. Parametrik çizim araçlarıyla zamandan tasarruf edinilebilinir. Parametrik çizim sayesinde tasarım revizyon süreniz önemli derecede azalacaktır. Nesneler arasında kalıcı ilişkiler tanımlayarak paralel çizgilerin paralel, eşmerkezli dairelerin merkezde kalmasını otomatik olarak sağlarsınız. Fikirlerinizi hayal edebileceğiniz istediğiniz her formda tasarlayabilirsiniz. Karmaşık nesneler oluşturmak, yüzeyleri daha yumuşak hale getirmek ve daha pek çok modelleme işi için yapmanız gereken tek şey yüzeyleri, kenarları ve köşeleri çekip-itmektir. Görselleştirme çalışmalarının bir adım ötesine geçer.

Yaptığınız iş son derece özel; kullandığınız yazılım da öyle olmalıdır. AutoCAD yazılımını özel ihtiyaçlarınıza uygun hale getirmek inanılmaz derecede kolaydır. Ayarlarınızı değiştirebilir, yazılımı genişletebilir, amacınıza uygun iş akışları kurabilir, kendi uygulamanızı geliştirebilir veya hazır bir tanesinden yararlanabilirsiniz [10].



Şekil 2.13 AutoCAD 2016 ekran görüntüsü [11]

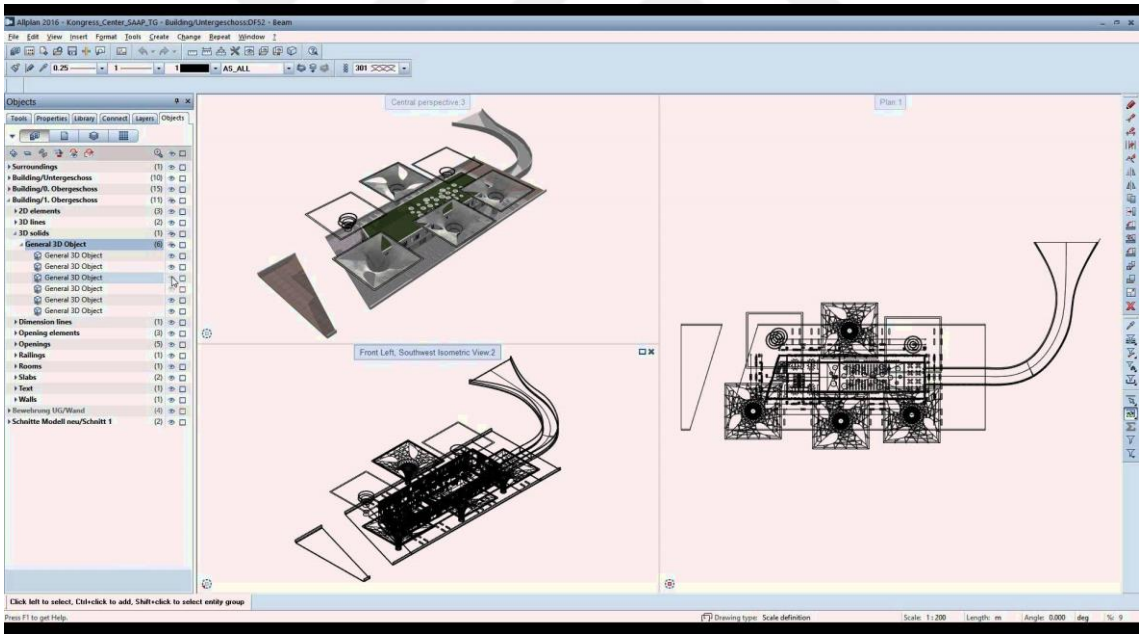
AutoCAD programı içerisine yama, plug-in ve programcıklar eklenebilir. Kendi komut satırı haricinde bir scripting özelliği bulunmamaktadır. Belli parametrik özellikler haricinde sizi özgür bırakan parametrik alt yapısı mevcut değildir

2.3.2.2 Allplan

Allplan Mimari eskizden etütlere, modellemeye, pafta düzenlemeye, detaylandırmaya, metrajlara ve uygulama çizimlerine kadar tüm tasarım süreçlerini kapsayan bir çözümdür. Bir model oluşturarak çalıştığınız için, bir projenin 2B, 3B veya her ikisini birlikte görebilme ve değiştirebilme yeteneği, çalışma sırasında eş zamanlı geri besleme sağlayarak tasarım süreçlerini hızlandırır. Allplan sadece tek bir yazılımda konsept tasarımından metraja kadar – sınırsız olasılıkları sunmaktadır.

Allplan'ın gelişmiş çizim araçları akıllı yakalama modları, dönen kürsörler, kolay kullanılabilir 2B geometrik çizim ve düzenleme komutları ve 2B elemanların hızla ve zahmetsizce 3B'ye dönüştürülmesi için yöntemler sunar. Allplan'ın akıllı ölçülendirme ve taramaları, tasarımdaki değişikliklere otomatik olarak uyum sağlar.

Allplan Mimari obje bazlı bir parametrik sistemdir. Örneğin, bir kapı, kendisinin bir kapı olduğunu "bilir"; kendi özelliklerini bilir, ancak aynı zamanda, bir duvara "ait olduğunu" ve bir boşluğa gereksinim duyduğunu da bilir. Eğer bir duvar eklemesi yapılırsa, üzerine eklendiği duvar otomatik olarak içerisine uyacağı bir boşluk oluşturur. Eğer kapı silinirse, duvar otomatik olarak boşluğunu kapatır. Uygulanan her değişiklikte, metrajlar, alanlar, hacimler ve yazıları, daima doğru kalacak şekilde güncellenirler. Allplan hızlı ve kolay tasarım için katların, döşemelerin, çatıların tanımlanmasında "referans düzlemleri" sistemini sunmaktadır. İlişkili görünümler modülünün en büyük avantajı, planda, cephede, kesitte veya perspektifte yapılan herhangi bir model değişikliğinin, bütün bu görünümlere otomatik ve eş zamanlı yansıtılmasıdır. Eğer, örneğin, bir pencereyi cephede taşırsanız veya planda bir kapı yerleştirirseniz, herşey, yapılan değişikliği yansıtmak için eş zamanlı güncellenir. İzometrik perspektiflerde de bu değişiklikleri uygulayabilirsiniz. Teknik bir çizim üzerinde çalışıyor olmanıza rağmen, 3B uzayda çalışmanın tüm avantajları kullanılmaktadır [12].



Şekil 2.14 Allplan ekran görüntüsü [13]

Allplan yukarıda da bahsedildiği gibi parametrik özellikleri olan bir programdır. Siz aslında 2 boyutta çalışırken projenizi 3 boyutlu hazırlarsınız. Tabi tüm bu özellikler Allplan'ın mevcut kütüphanesi aracılığı ile olmaktadır. Her ne kadar kütüphanesini istediğimiz şekilde genişletebiliyor olsakta içerisinde scripting özelliği bulunmadığı için sürekli belli sınırlar içerisinde kalmaktadır.

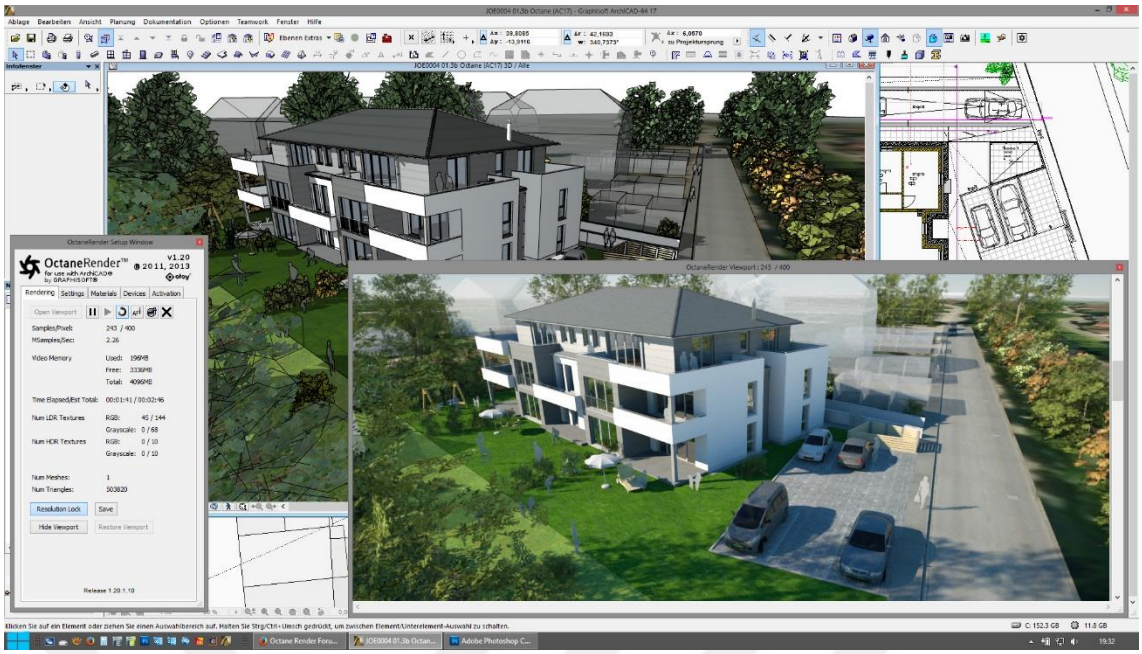
2.3.2.3 ArchiCAD

1988 yılında yapı sektörü ve mimarlar klasik CAD programlarından farklı bir yazılım olan ArchiCAD'le tanıştılar. Çizgiler, taramalar ve yazılardan oluşan çizimin ötesinde, tasarım yapabilmeye, yapı bilgi bankalarını oluşturup kullanabilmeye başladılar. Böylece, ArchiCAD kullanıcıları bilgi çağından yararlanan ilk mimarlar oldular.

ArchiCAD iş akışı sürecinize farklı bir yaklaşım sunarak belgelemenizde doğruluğu ve verimliliği korurken tasarımınız üstünde daha fazla kontrol sağlar. Duvarları yükseltirken, döşemeleri yerleştirirken, kapılar ve pencereler eklerken, merdivenler oluştururken ve çatılar inşa ederken Bina Bilgisi Yazım Aracı 3 boyutlu model bilgilerinden oluşan bir veri tabanı oluşturur. Bu araçtan tasarımınızı tam olarak tanımlamak için gerekli tüm bilgileri çıkarabilirsiniz. Tam bir plan, kesitler ve cepheler, mimari ve uygulama detayları, Keşif Özetleri pencere/kapı/bitirme planları, görsel sunumlar, animasyonlar ve sanal gerçeklik sahneler. Bu siz tasarım yaparken ArchiCAD'ın size çok az tekrar tekrar yapılacak can sıkıcı çizim işleri bırakarak tüm projelerin belgelerini yaratacağı anlamına gelir. Ve bu aynı zamanda 2 boyutlu yazılımda tasarım yapmanın aksine Virtual Building™ yaklaşımının masraflı hatalara veya üretkenliğinizi kaybetmenize neden olmadan belgelerinizin bütünlüğünü korurken değişiklikler yapma şansı yarattığını da ifade eder.

Danışmanlar hangi CAD (Bilgisayar Destekli Tasarım) platformunda oldukları fark etmeksizin bina verisini elektronik formatta alabilir, üstlerinde değişiklikler yapabilir ve bu süreç içinde hiç veri kaybolmadan dosyayı daha ileri çalışmalar için size geri verebilirler. Kapsamlı programlar ve malzeme listeleri hassas ölçekli detayların çizimleriyle birlikte inşaatçılar ve alt yükleniciler için hazır olur. İnşaatçılar binanın inşaatının veya yıkımının her aşamasında işlerini planlayabilir, zamana dayalı animasyonlar ve belgeler yaratabilir.

Her mimar veya iç mimar ArchiCAD kullanarak, günümüz bilgi çağının olanaklarını kolayca elde edebilir ve bilgi toplumunun bir üyesi olabilir. Çünkü ArchiCAD hem Macintosh, hem de tüm Windows platformlarında mimarlara ve yapı sektörüne en ileri bilgi işlem teknolojilerini kendine özgü kolaylıklarla sunuyor [14].



Şekil 2.15 Archicad ekran görüntüsü [15]

ArchiCAD, AllPlan ile çok benzer özelliklere sahip olan ve onunla aynı platformda yer alan bir program. Dolayısı ile scripting özelliğinin olmadığını ve sizi sınırlayan bir kütüphanesi olduğunu söyleyebiliriz.

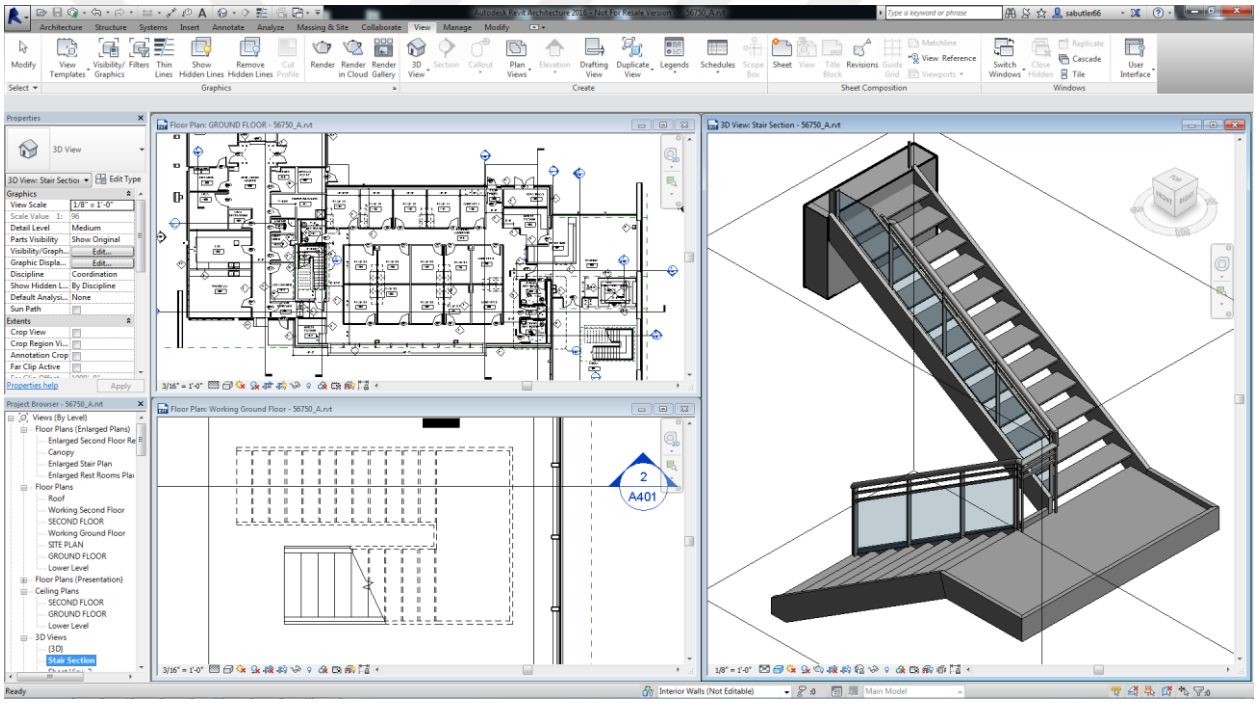
2.3.2.4 Revit

Revit, gerçek anlamda Parametrik Yapı Bilgi Sistemi içeren bir yazılım. Arazi ve kütle çalışmasından detaylı uygulama çizimlerine ve metraja kadar, tüm aşamalarda üretkenliği, koordinasyonu ve kaliteyi artıran araçlar içeren Revit, veritabanı bazlı ve parametrik çalışan bir yazılımdır. Akıllı nesneler kullanarak her türlü yapıyı en ince detayına kadar 3 boyutlu olarak ve kısa sürede çizmenizi sağlayan araçlar içeriyor. Çizimler aynı veri tabanında ilişkili olarak çizildiğinden, planlar çizilirken kesit, görünüş ve 3. boyutta bütün bina ve ayrıca Metraj da hazırlanıp çizim, ek yazılıma ihtiyaç duyulmadan sonuçlandırılabiliyor. Revit'in parametrik yapısı, herhangi bir anda yapılan her değişikliği (görünüşte, kesitte, metraj listesinde, paftada, kısacası nerede yapılırsa yapılsın), tüm dokümanlara otomatik olarak iletir.

Tasarım, çizim, görselleştirme ve metrajı bir arada içermesi verimliliğin artmasını sağladığı için Mimarlar arasında yaygınlaşan Revit, binaya tesisat projesi çizmeyi

sağlayan Revit MEP ile de uyumlu çalışıyor. Yeni mental ray render motoruna da sahip olması ile fotorealistik görseller konusunda da oldukça başarılı [16].

Büyük projelerde, farklı Revit modelleri birbirine bağlanabilir. Her model üzerinde bağımsız çalışılabildiği gibi, diğer modellerin bilgileri referans ve sunum için kullanılabilir. Revit, tek başına kullanılabileceği gibi, Autodesk'in diğer tasarım yazılımları ile birlikte de kullanılabilir. DWG ya da FBX aracılığıyla aktarılan modeller, çok gelişkin görselleştirmeler için Autodesk VIZ içersine alınabilir. DWG, DXF ve DGN formatlarını ve ACIS katılarını destekler. AutoCAD ile veri paylaşımında sorun yaşanmadığı için, AutoCAD kullanılan diğer disiplinler (statik, elektrik, tesisat, vb.) ile çizim paylaşımı olanaklıdır. Revit Architecture'un Yapı Bilgi Sistemi'ndeki veriler, daha detaylı maliyet, planlama, vb. işler için ODBC uyumlu veritabanlarına aktarılabilir [36].



Şekil 2.16 Revit 2016 ekran görüntüsü [17]

Revit, AllPlan ve ArchiCAD ile aynı platformdadır ve çok benzer özelliklere sahiptir. Diğerlerinden artı özelliği Autodesk ürünü olduğu için 3ds Max ve AutoCAD programlarıyla çok daha sağlam bir bağlantı içerisindedir.

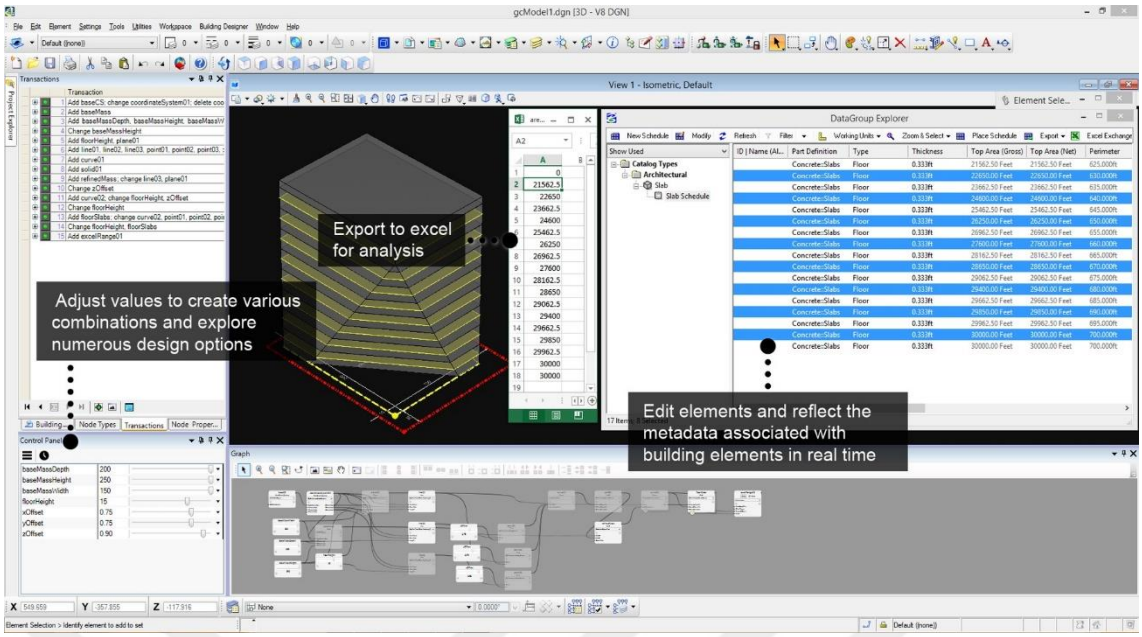
Yukarıda bahsettiğim programlar (AutoCAD, Allplan, ArchiCAD ve Revit) parametrik bir yapıya sahip olan ve kullanıcıya belirli şablonlar sunan başlıca programlardır. Bu programlar kullanıcıya en sık kullanılan şablonları sunar ve aslında yapı elemanlarının

birbirleriyle ilişkilendirildiği parametrik yapıya sahiptirler. Scripting özellikleri yoktur. Yani kullanıcı kısıtlı özgürlüğe sahiptir. Kendi arayüzünü ya da kendi çizgilerini oluştururken programın olanaklarıyla sınırlıdırlar. Asıl hedefleri kullanıcıya özgür tasarım alanı yaratmaktan ziyade çizim yükünü hafifletmek amacı gütmektedirler. Tasarım ve görselleştime özelliği daha yüksek olan ve scripting özelliği barındıran diğer programları da inceleyelim.

2.3.2.5 Generative Components

Generative Components, Bentley Systems tarafından ilk kez 2003'te tanıtılan, 2005 yılında kullanımı ciddi oranda artmaya başlayan ve 2007 yılında ticari olarak piyasaya sürülen parametrik bir CAD yazılımıdır. Generative Components kısaca GC olarak anılır. GC yazılımının temel kullanıcılarının geleneksel tabanını akademik ve teknolojik olarak gelişmiş tasarım firmaları oluşturmaktadır. 3D katı modellemeden daha fazla akıcılı ve akışkanlık sağlamak isteyen, 3 boyutlu parametrik modelleme yeteneklerine sahip bir program olarak özetleyebiliriz.

Bu yazılım aracılığıyla dinamik modeller üretilebilmekte ve geometri doğrudan manipüle edilebilmektedir. Uygulanan kurallar modelin elemanları arasındaki dinamik ilişkileri kurmakta, karmaşık formları ve sistemleri tanımlarken ise algoritmalar kullanılmaktadır. Yazılım Bentley Systems için DGN, AutoDesk için DWG, Rhino için STL(Stereo Lithography) gibi birçok endüstriyel standartlardaki giriş-çıkış formatlarını desteklemektedir. Bu sayede BIM sistemi ile uyumlu bir şekilde çalışabilmektedir. GC CAD yazılımı öncelikle mimarlar ve mühendisler tarafından bina tasarımları için kullanılmıştır. Fakat doğal ve biyolojik yapıların ve matematiksel sistemlerin modellenmesi içinde kullanılmıştır [18].



Şekil 2.17 Generative Components ekran görüntüsü [19]

Diğer programların aksine, çizgisel ya da katı model olarak çalışmak yerine scripting mantığı ile çalışmakta ve tamamiyle parametrik bir alt yapıya sahiptir. Görselleştirme konusunda kaygısı yoktur. GC ile yapının parametrik modeli oluşturulabilmektedir. Üç boyutlu modellerin her bir bileşeni diğer bileşenle ilişkili olarak oluşturulabilmekte ve sayısal değerleri birbirleriyle ilişkili olarak değiştirilebilmektedir.

2.3.2.6 Lumion

Lumion, bina tasarımlarını görselleştirmek için kullanılan bir programdır. Etkileyici, yüksek kalitede videoları çok çabuk oluşturur ve öğrenilmesi son derece kolaydır. Lumion'ın 90'ın üzerinde ülkede kullanıcıları bulunmaktadır. Kullanıcıları arasında mimarlar, tasarımcılar, mühendisler, BIM modelleyicileri ve üniversiteler bulunmaktadır. Lumion, 3B görselleştirme yapım sürecine yeni bir tanım getirmiştir. Birçok AEC profesyoneli için BIM araçlarının önemli bir parçasını haline de gelmiştir.

Lumion, zaman içerisinde modelinizde birçok kere değişiklik yaptığınızda çok başarılı oluyor. Modelinizi programın içine aldıktan ve Lumion sahnenizi oluşturduktan sonra bile, tek bir tıklama ile modellerinizi güncelleyebilirsiniz. Güncellenmiş model dosyanızı programa aldığınızda Lumion yaptığınız bütün değişiklikleri tekrar uygulayacaktır [20].



Şekil 2.18 Lumion Ekran Görüntüsü [21]

2.3.2.7 Unity

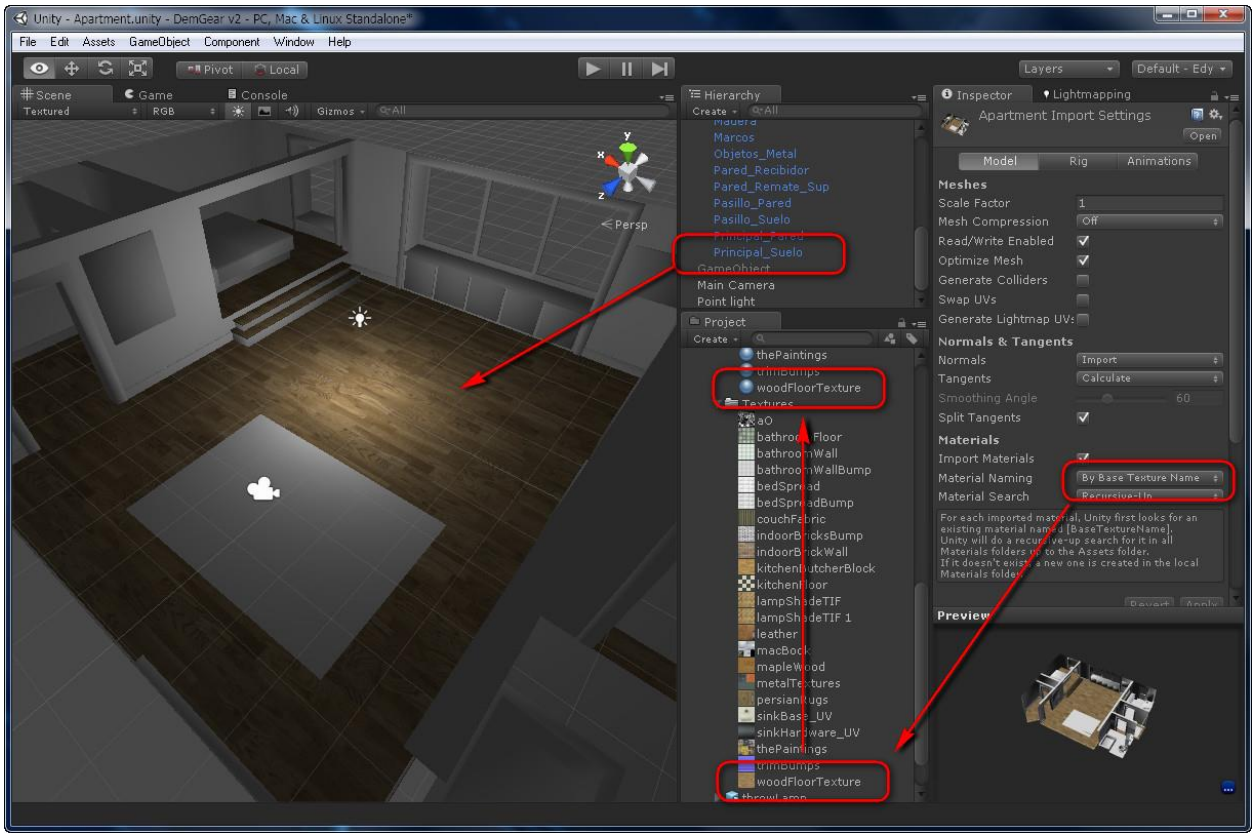
Unity 3D, bir oyun motorudur. Oyun ve bilişim dünyasına getirdiği en önemli yenilik, gelişmiş özelliklere sahip 3 boyutlu oyunların bilgisayara kurulmadan oynanmasını sağlamak olmuştur. Unity 3D motorunu kullanan oyunlar, Unity Web Player eklentisi sayesinde hiçbir kurulum işlemi olmadan web tarayıcı üzerinden çalışabilmektedir. Bu yönü sayesinde oyunlar bilgisayara daha az yüklenirken bir oyuncuyu yeni bilgisayar alma zahmetinden kurtarmış olur.

Unity'nin oyun yapımcılarına sağladığı başka bir kolaylık da Unity ile geliştirilen bir oyunun herhangi bir altyapı değişikliğine gerek olmadan farklı platformlara (PC, Mac, Web, iOS, Android, Windows Phone) uygun olarak derlenebilmesidir. Bu sayede PC için hazırlanan bir oyun tek tıklamayla Mac içinde çalışır hale getirilebilir.

Unity son derece pahalı diğer gelişmiş oyun motorlarının (Havok, vb.) sunduğu gelişmiş shader yazılımı, fizik motoru, animasyon editörü, occlusion culling gibi özellikleri, uygulama ve oyun geliştiricilerine makul bir fiyattan sunmaktadır.

Unity'nin diğer oyun motorlarından üstün taraflarından biri de oyun geliştirme zamanında geliştiriciye program kodu yazma olanağı vermesidir. Diğer oyun

motorlarının ekserisi grafik ile kodu ayırmışken, Unity ile grafik ve kod birlikte çalışmaktadır. Bu çalışma mantığı geliştiriciye esneklik sağlamakta, geliştirme süresini kısaltmaktadır.



Şekil 2.19 Unity ekran görüntüsü [22]

Unity bütün bu avantajlarının yanında Unity 3D'de yazılmış oyunlar düşük ve orta seviye bilgisayarlarda rahatlıkla kullanılabilmektedir.

Unity oyun hazırlama yanında, mimari ve eğitim amaçlı simülasyonlar hazırlamaktada kullanılabilmektedir. En son versiyonu ile kişisel (Personal) kullanımlar için ücretsiz olan yazılım sayesinde daha geniş bir kitleye hitap edecek şekilde bir pazarlama taktiğine yönelmiştir [23].

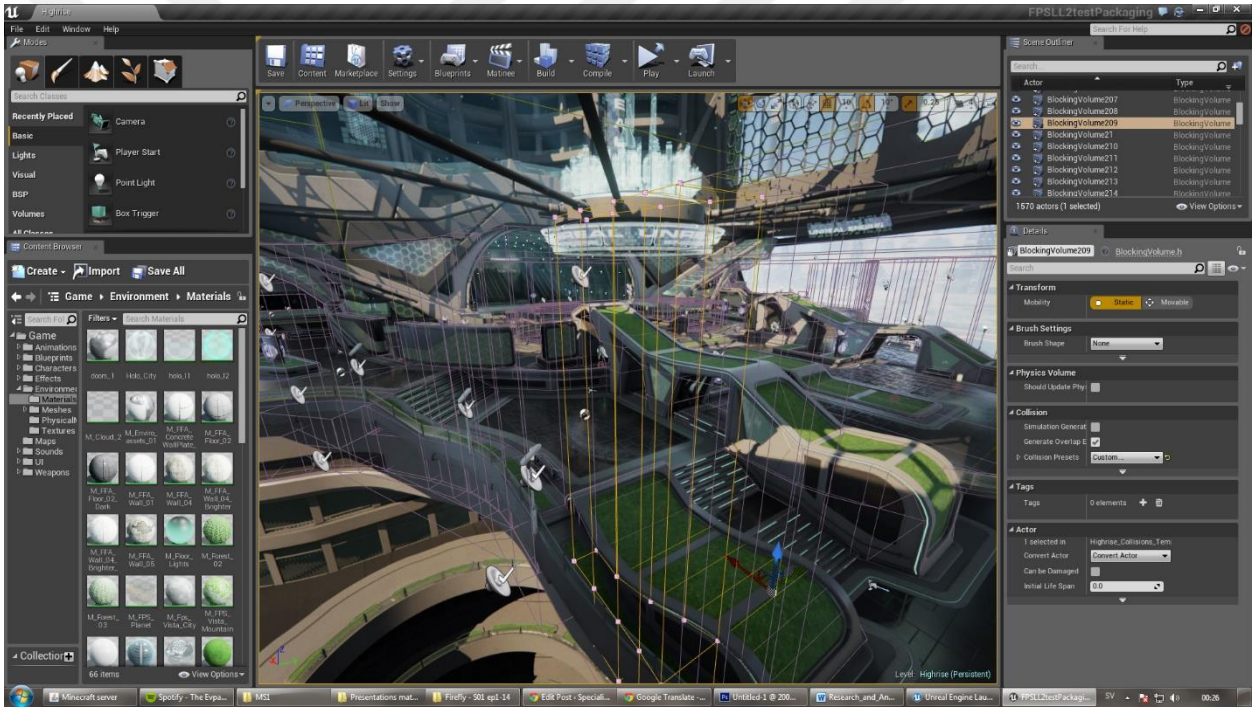
2.3.2.8 Unreal Engine

Unreal Engine, Epic Games tarafından geliştirilen ve ilk olarak 1998 yılında yayınlanan Unreal isimli birinci şahıs nişancı oyununda kullanılan oyun motoru. Esas olarak birinci şahıs nişancı oyunları için geliştirilmesine karşın, sonraları çeşitli türlerdeki oyunlarda

kullanıldı. C++ ile yazılmış kodu sayesinde yüksek derecede taşınabilirliğe sahiptir ve günümüzde birçok oyun geliştirici tarafından kullanılan bir araç haline gelmiştir.

Şu anki sürümü Unreal Engine 4'dur, Microsoft Windows, Xbox One, Windows RT için DirectX; OS X, Linux, PlayStation 4, iOS, Android için OpenGL ve internet tarayıcılar için JavaScript/WebGL kullanır [24].

Yeni bilgisayar teknolojilerine uyum, içerik yaratım araçları ve teknik destek altyapısı sayesinde motor tüm oyun gelistiricileri tarafından tercih edilmektedir. Unreal Engine kullanılarak hazırlanan oyun uygulamaları düsük bir maliyet ile farklı bilgisayar ortamlara da aktarılabilmektedir. Bu durum hazırlanan uygulamaların kullanım yaygınlıgını arttırmaktadır [25].



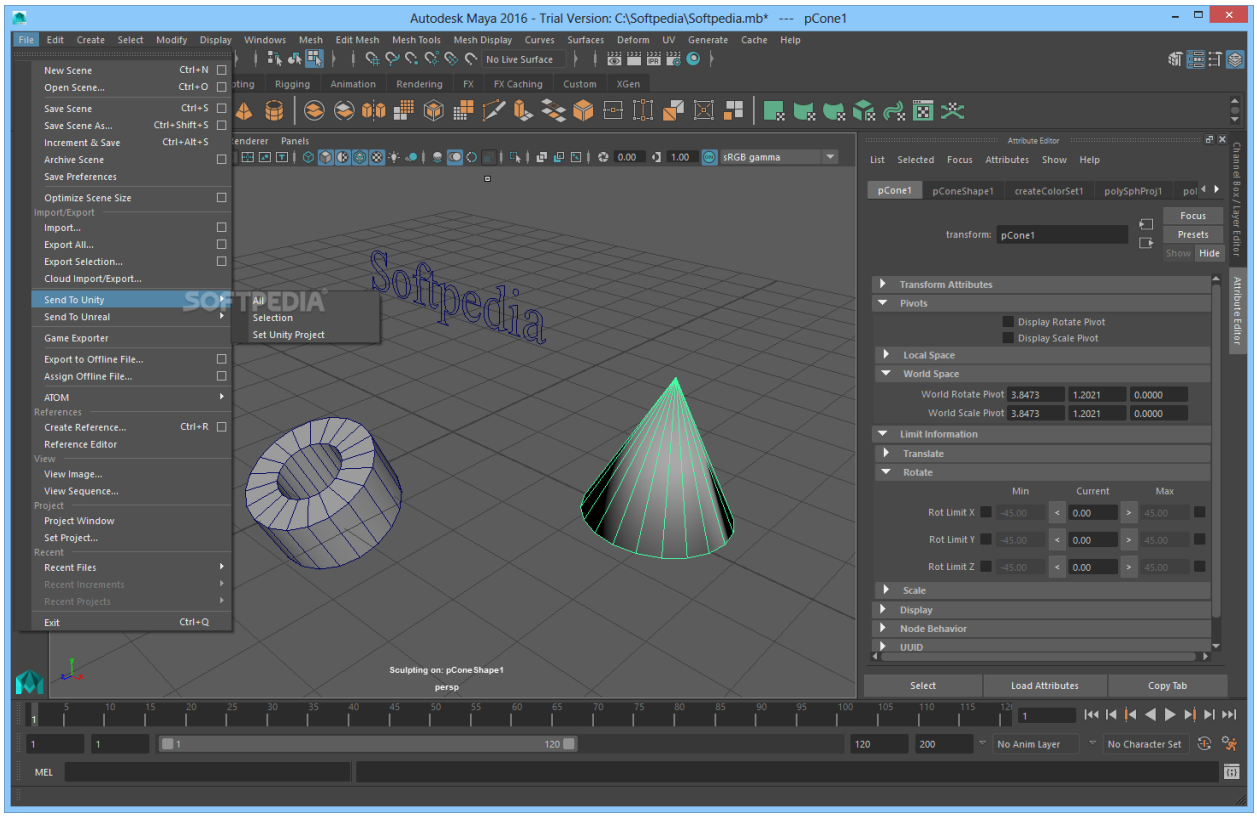
Şekil 2.20 Unreal Engine ekran görüntüsü [26]

2.3.2.9 Maya (MelScript)

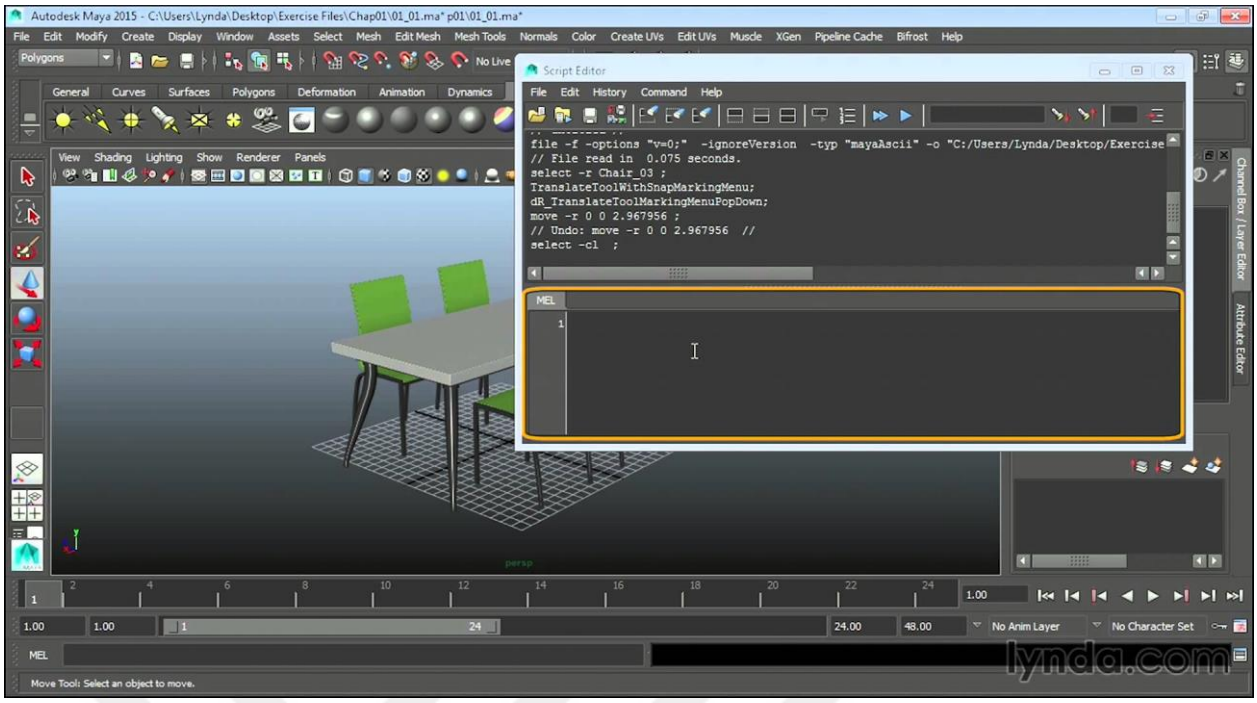
Maya, Alias firmasınca uzun yıllar çıkarıldıktan sonra 2006 yılında 7.0 sürümü sonrası Alias firmasına ait diğer tüm yazılım ürünleriyle birlikte Autodesk firmasınca satın alınan komple çözüm sunan bir animasyon paketidir. Tasarım ve görsel efekt iş ve işlemleriyle uğraşan tüm dünya profesyonellerinin tercih ettiği bir yazılım olmasıyla da bilinir. Tercih edilmesinin bir sebebi de, açık script yapısıdır. Profesyoneller genelde projeye göre

hazırladıkları script ve pluginleri Maya'nın açık yapısı sayesinde çok kolayca kullanabilmektedirler [27].

"Bilim ve teknoloji dalında Oskar ödülü" kazanmış Autodesk Maya yazılımı çok güçlü araçlarıyla 3B modelleme, karakter canlandırma, görsel efekt üretimi ve kaplama yapılabilen ve sinema-reklam sektöründe yoğun olarak kullanılan bir yazılımıdır. Açık yazılım mimarisi ile her işe yönelik kodlamalar yapabilirsiniz veya bütünleşik programlayıcıyı kullanarak ihtiyaç duyduğunuz eklentilerin geliştirilmesine olanak sağlar. Geniş dokümantasyon ve kapsamlı (API) yazılım geliştirme ara yüzü sayesinde yaratıcı fikirlerinizi gerçekleştirebilirsiniz.



Şekil 2.21 Maya 2016 ekran görüntüsü



Şekil 2.22 Maya Mel-Script ekran görüntüsü

Autodesk Maya video ve film üretimi yapan profesyonellerin tüm ihtiyaçlarını karşılar. Dünya çapında oyun geliştiricileri, tasarım görselleştirme, gerçekçi canlandırma ve sıra dışı görsel efektler oluşturma, fotoğraf gerçekliğinde dijital resimler yaratan profesyoneller sanatçılar tarafından yaygın şekilde kullanılmaktadır.

Autodesk çözümleri dijital medya yaratma işinizdeki her konuda işbirliği araçları ve etkin takım çalışması olanakları sunarak hızınızı ve tasarım sürecindeki verimliliğinizin artırır. Autodesk 3ds Max yazılımı ile karşılıklı karmaşık veri değişimi işlemleri geliştirilmiş FBX teknolojisi ve değiş tokuş edilebilir geometri önbelleği sayesinde birlikte çalışabilirlik imkanı sağlar. Yeniden kaplama işlemine gerek kalmadan Autodesk Toxik kullanan post-production tasarımcıları ile sorunsuz biçimde işbirliği yapar beraber çalışır. Yarattığınız sahnelerin hedef kaplayıcıda nasıl görüneceğine doğrudan Maya içinde karar verebilirsiniz [28].

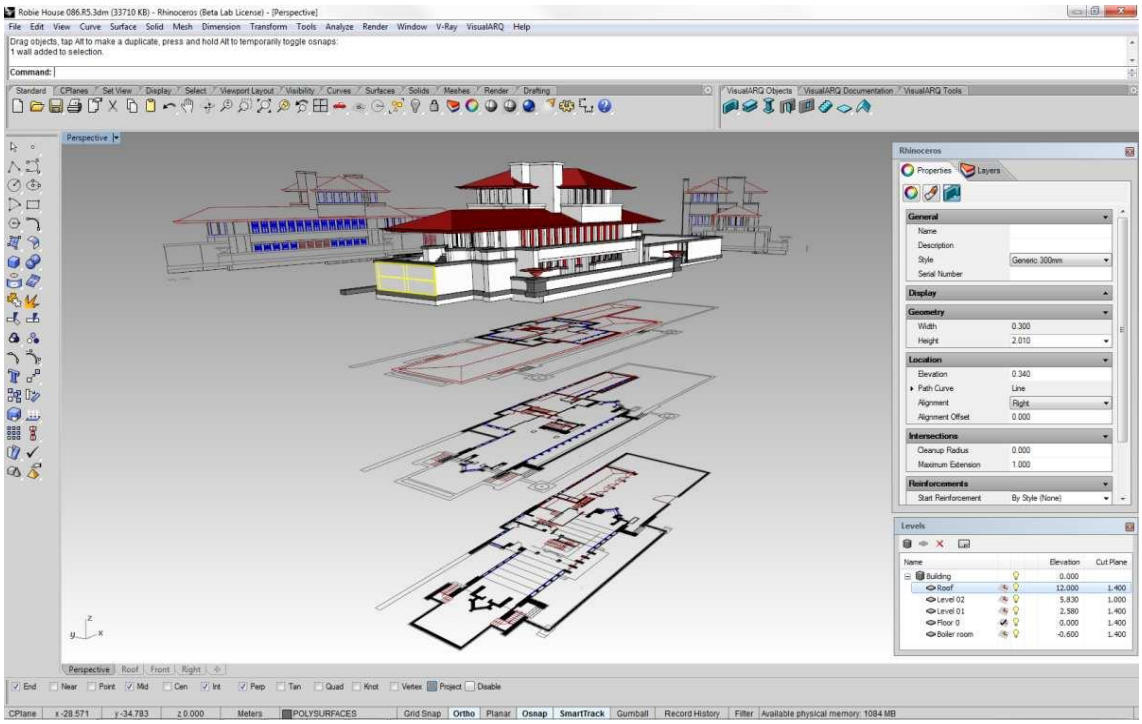
2.3.2.10 Rhinoceros (RhinoScript - Grasshopper)

Rhinoceros 3D, özellikle sanayi için 3B modelleme ve prototipleme için tasarlanmış bir CAD/CAM yazılımıdır. Modellemede Kafes (Mesh) modelleme yerine NURBS modelleme kullanıldığı için, 3B prototipleme için uygundur.

Rhinoceros 3D, NURBS tabanlı ticari bir modelleme programıdır. İlk olarak Robert McNeel & Associates şirketi tarafından, Autodesk şirketinin Autocad programı için geliştirilmiş bir eklenti olarak ortaya çıkması düşünülen program daha sonra bundan vazgeçerek kendi adında özgün bir yazılım olarak gelişmiştir. Genellikle endüstri tasarımı, mimarlık, deniz araçları tasarımı, takı tasarımı, otomotiv tasarımı, CAD/CAM, seri üretim, tersine mühendislik ve multimedya ve grafik tasarım alanlarında yaygın olarak kullanılmaktadır.

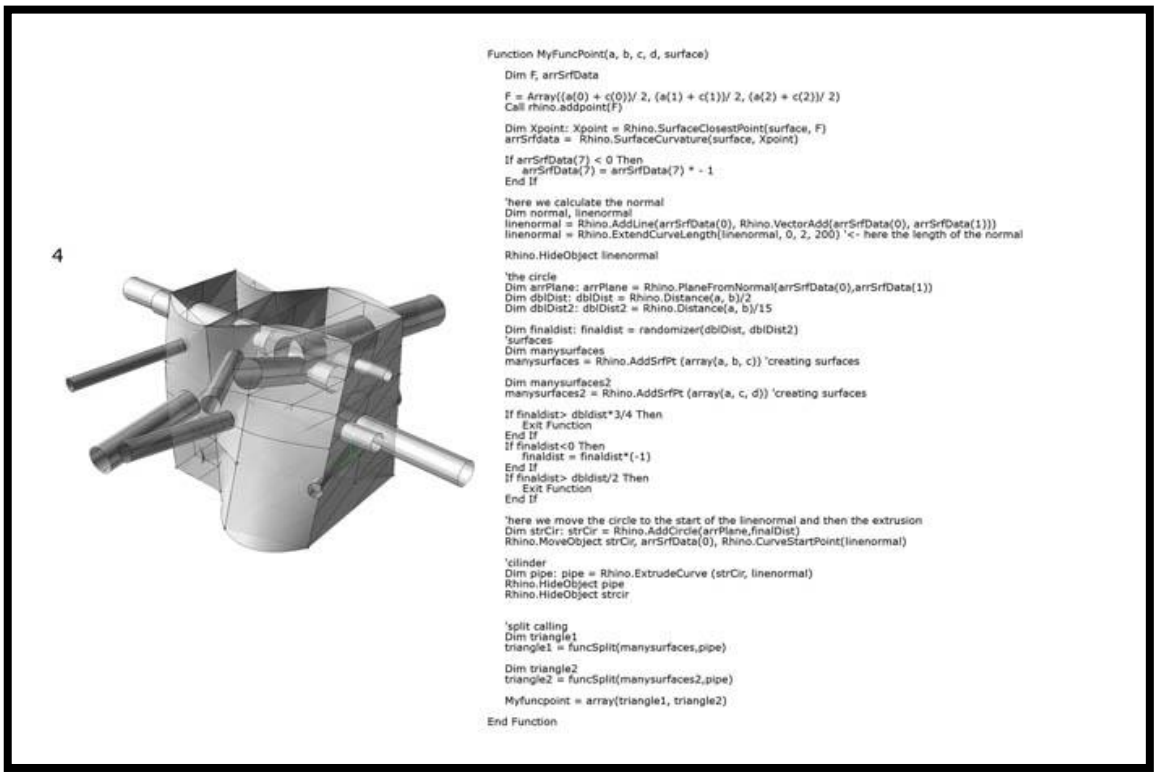
NURBS tabanlı olması dolayısıyla özellikle sanayi üretiminde yaygın olarak kullanılmaktadır. Mesh (ağ) tabanlı olan 3ds max gibi programlardan bu yönüyle ayrılır. Çok yüksek bir hassasiyete sahiptir.

Rhino 3D birçok farklı endüstri tarafından çeşitliliği, kullanışlı komutları ve düşük maliyeti nedeniyle tercih edilmektedir. Çok çeşitli import (içeri alma) ve export (dışarı verme) özelliği sayesinde diğer programlar arasında bilgi alışverişi nedeniyle de beğenilmektedir. Rhino'nun birçok farklı çeşitte dosyaya destek vermesi programlar arasındaki açık bulunan iş akışını hızlandırır. Rhino 3d özellikle serbest formlu NURBS modelleme de uzmanlaşmıştırRhino 3D'nin, Visual Basic tabanlı RhinoScript isminde bir de kodlama programı vardır [29].

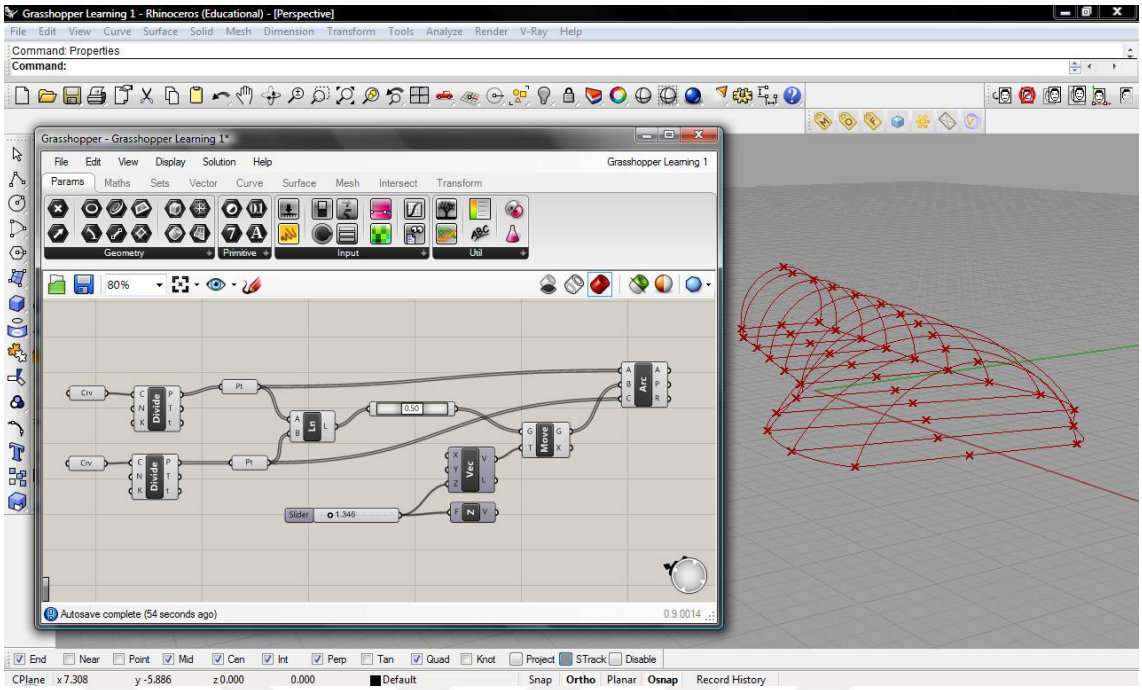


Şekil 2.23 Rhinoceros ekran görüntüsü [30]

RhinoScript, kuyumculuk (takı tasarımı), gemi mühendisliği, uçak mühendisliği, kalıp çekirdek parça modelleme, animasyon karakterleri modelleme, futbol oyunları için stadyum yaması yapma gibi pek çok alanda kendisini göstermiştir. Bunun en büyük etkeni kullanıcı dostu yapısı ve scripting özelliğidir. Son yıllardaki parametrik mimarlığın gelişimi mimarların da Rhinoscripti kullanmalarında önemli rolü olmuştur. Özellikle pek kullanıcının tercih ettiği render motoru Vray'in de rhinoscript aracılığı ile Rhinoceros içerinde yer almasından sonra geç de olsa ülkemizde de yaygınlaşmaya başlamıştır.



Şekil 2.24 RhinoScript ekran görüntüsü [31]



Şekil 2.25 Grasshopper ekran görüntüsü [32]

Bu gelişmelere ilaveten Rhinoceros içerisine Grashopper eklentisi de gelmiştir. Grasshopper, Rhino üzerinde çalışan ve rhino history bilgisini kullanarak algoritmik modelleme imkanı sunan bir eklentidir. Hiçbir programlama bilgisi olmadan da kullanılabilir. Grasshopper arayüzü ise buna imkan sağlamaktadır. Grasshopper'da modelleme parçaları kullanılarak modelleme yapılır ve model oluşturulduktan sonra her bir parçanın kendi ayarları değiştirilerek model tekrar oluşturulabilir [33].

2.3.2.11 3DS Max (MaxScript)

3ds Max, Autodesk tarafından geliştirilen bir 3D modelleme, görselleştirme ve animasyon programıdır. MSDOS ortamında çalışan 3D Studio yazılımının devamı olan 3ds Max'in son sürümü, 2015 yılında çıkan 3ds Max 2016'dir.

Gelişmiş eklenti desteği ve kolay kullanımı ile 3ds Max, 3D modelleme yazılımları arasında en yaygın kullanıma sahip uygulamalardan biridir. Gelişmiş karakter modelleme özellikleri ile oyun geliştiricilerinin gözdesi haline gelmiştir. Film özel efektleri, mimari sunumlar ve endüstriyel tasarım sunumları gibi alanlarda da yaygın olarak kullanılmaktadır.

DOS ortamında çalışan ilk sürümü Yost Group tarafından üretilmiş ve Autodesk tarafından yayımlanmıştır. 3D Studio 4. sürümünden sonra Windows NT platformu için yeniden yazılmış ve "3D Studio Max." adını almıştır. Bu versiyon ise Kinetix tarafından piyasaya sürüldü, Kinetix, Autodesk'in alt kuruluşu olan Medya ve Eğlence bölümüydü. Yazılım daha sonra geliştirilmeye devam etti ve ismi kısaltılarak "3ds Max" olarak değiştirildi.

3ds Max çok sayıda temel nesneyi hazır olarak sunar. Mimari tasarımlar için de duvar, kapı, pencere ve merdiven gibi bileşenleri ölçülerini kolayca değiştirerek projeye eklemek mümkündür. 3ds Max ayrıca poligonal modelleme, NURBS modelleme, yüzey modelleme gibi teknikleri destekler.

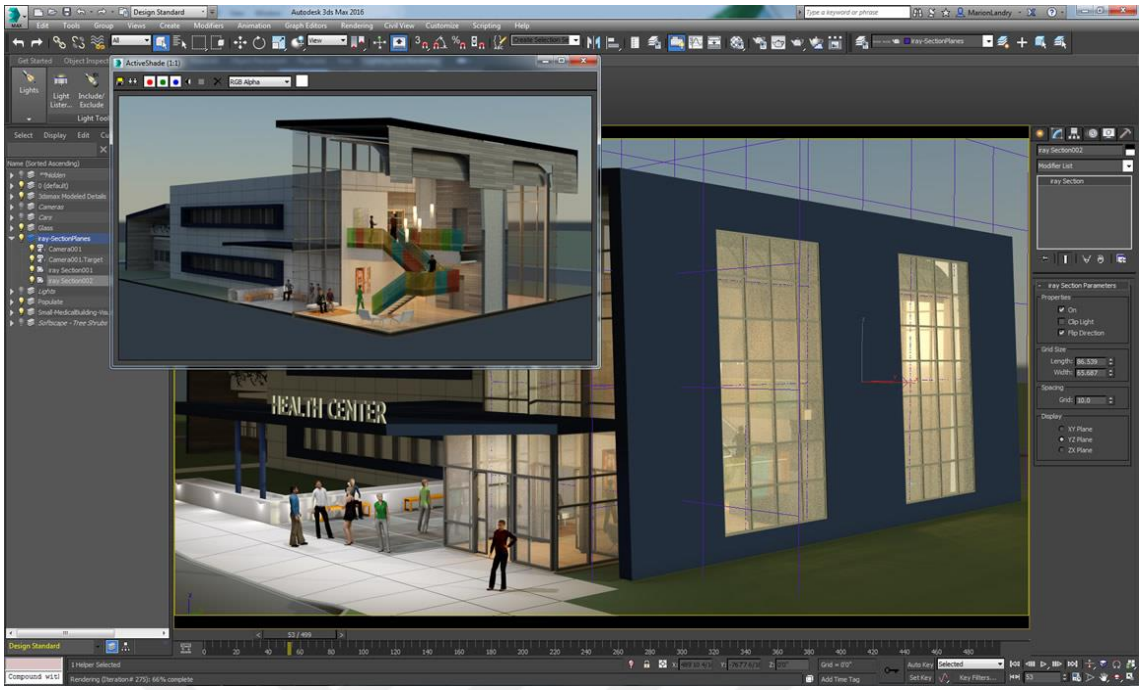
3ds Max'in animasyon kontrolleri ile nesnelerin tüm özellikleri, materyaller, kameralar, ışıklar ve çevre özellikleri zaman içinde değiştirilebilir ve Curves Editor ile tüm bu özellikler üzerinde tam bir kontrol sağlanabilir. Değişken grafiklerinin Bezier eğrileriyle kontrol edilebildiği bu editör ile karmaşık animasyonların üstesinden gelmek mümkündür.

3ds Max, animasyon için klasik anahtar kare yöntemini kullanır. Zaman doğrusu içinde farklı noktalarda verilen değerler arası geçişi otomatik olarak yapar ve Curves Editor ile bu geçişlere ince ayarlar yapılmasına olanak verir.

Ters Kinematik çözümleyicisi ile birbirleriyle bağlantılı hareket eden objeler arası ilişkiler kolayca çözümlenir ve kare anahtarlama yöntemi ile kompleks mekanizmaların animasyonu yapılabilir. Ayrıca, pozisyon, bakış, yüzey, bağlanma, tutunma ve yönelme kısıtlayıcılarıyla gelişmiş animasyonlar yapılabilir.

Animasyon için kullanılabilecek diğer özellikleriyse uzay saptırıcıları ve niteleyicilerdir. Uzay saptırıcıları kendilerine bağlanan objelere, bükme, patlatma, rüzgar ve yerçekimi gibi etkileri uygularlar. Niteleyiciler ise modellemede kullanıldıkları gibi animasyon için de objeleri zaman içinde değiştirmede kullanılabilirler.

3ds Max'in en güçlü özelliklerinden biri de Havoc tarafından yazılan ve pek çok oyunda kullanılan ünlü fizik motoru reactor'dür. reactor ile yerçekimi etkisiyle düşme, esneme, sıçrama gibi fizik benzetimleri yapılabilir [34].



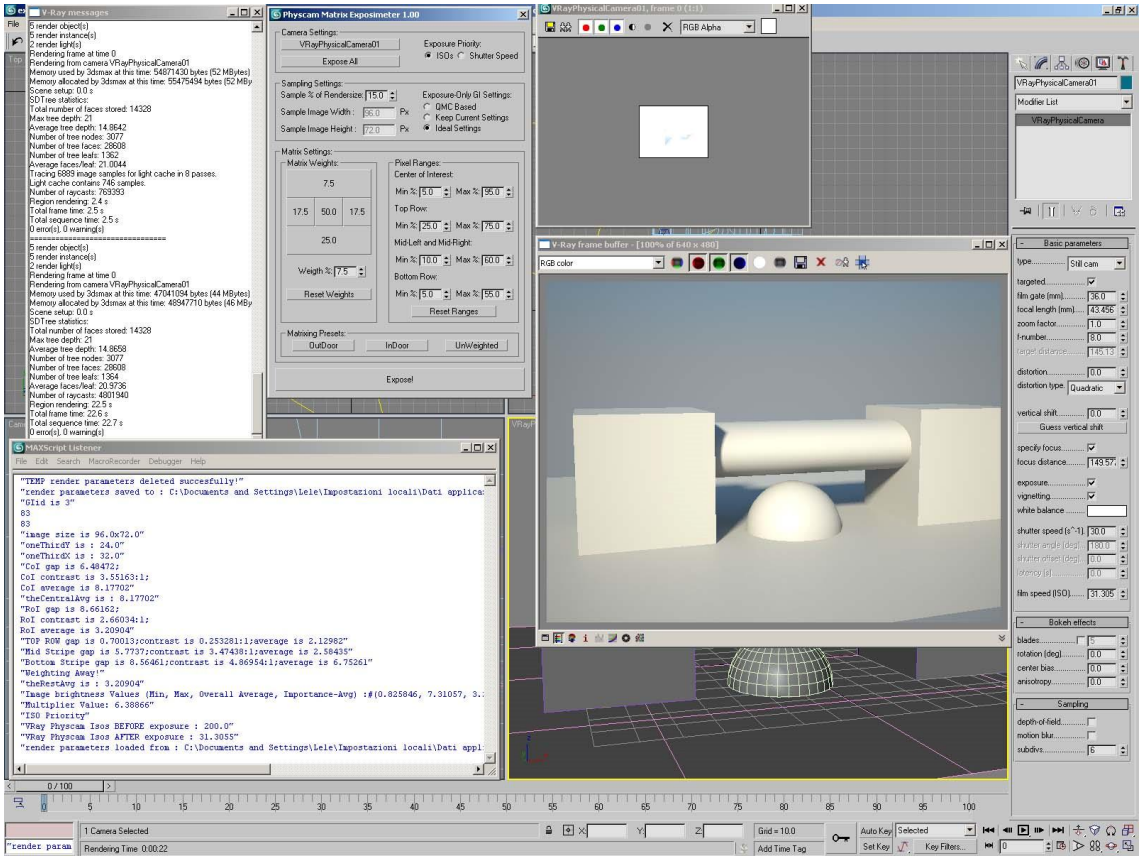
Şekil 2.26 3DS Max 2016 ekran görüntüsü [35]

3dsmax 2009 design'la başlayan Autodesk Revit uyumluluğu ile de bir paket programın bir nevi disiplinler arası çalışması sağlanmıştır. Recognize sahne yükleme teknolojisi 3ds Max Design ve Revit Architecture yazılımı arasında yeni ve yüksek seviyede bir uyumluluk ve birlikte çalışabilme olanağı sağlar. FBX tabanlı Recognize araç seti size Revit Architecture 2009 içindeki projenizi 3B model, ışıklar, kameralar ve malzeme tanımlamaları ile eksiksiz ve doğru bir şekilde 3ds Max Design 2009'a almanızı sağlar. YBS verilerinizi güçlü bir görsel iletişim aracı olarak kullanabilmeniz sağlanır.(19) Güneş, gök ışığı ve yapay ışıklandırma analizleri yapmak ve ışığın tasarımla nasıl bir etkileşim içine gireceğini doğru biçimde tahmin etmek suretiyle daha sürdürülebilir tasarımlar gerçekleşmektedir. Radiance'ı gün ışığı simülasyon aracı olarak onaylayan NCR, 2010 versiyonu ile de Exposure'u da onaylamıştır.

Tüm bu özellikleri ile 3dsmax programı tasarım süreci içinde daha çok görselleştirme adımında programa verilen plugin desteği ile kendini göstermiş ve kullanıcılarının daha çok bu yüzden programı seçmesini sağlamıştır [36].

3ds Max, parçacık sistemleri, karakter modelleme araçları, hareket yakalama araçları ve gelişmiş denetçiler gibi özellikleriyle tek bir pakette çok sayıda özelliği sunmaktadır.

Ayrıca MAXScript adında tümleşik bir programlama dili vardır. Bu programlama dili sayesinde max sahnesinde yapabildiğiniz hemen her şeyi, hatta çok daha fazlasını Maxscript kullanarak yapabiliriz. Üstelik kendi arayüzünüzü oluşturabilir ve yapmak istediğiniz, belki saatler sürecek olan bir işlemi birkaç tıklama ile halledebilecek kadar çabuk sonuca ulaşabilirsiniz.



Şekil 2.27 MaxScript kullanılmakta olan bir 3DS Max ekran görüntüsü [37]

MaxScript, RhinoScript ve MelScript hemen hemen benzer özelliklere ve benzer olanaklara sahiptirler. Burada MaxScript'in 3 özelliği ön plana çıkıyor.

1. RhinoScipt ve MelScript'ten biraz daha basit bir kod mantığı olması.
2. İçerisinde çalıştığı 3DS Max'in çok daha fazla kullanıcısının olması.
3. 3DS Max'in daha fazla plug-in ve görselleştirme desteğine sahip olması.

Çizelge 2.1 İncelenen programlar ve işlev çizelgesi

Sınıfı	Program	2D	3D	Animasyon	Parametrik Yapı	Scripting	Rendering
CAD	AutoCad	+	+	+	/	-	/
BIM	AllPlan	+	+	/	+	-	/
	Revit	+	+	/	+	-	/
	ArchiCad	+	+	/	+	-	/
Simülasyon	Unity3d	-	+	+	-	+	+
	Unreal Engine	-	+	+	-	+	+
	Lumion	-	+	+	-	-	+
Modelleme	Gen. Comp.	-	+	/	+	+	/
	Sketch-Up	+	+	/	-	-	+
	Rhinocerous	+	+	+	+	+	+
	Maya	+	+	+	+	+	+
	3ds Max	+	+	+	+	+	+

+	Var
-	Yok
/	Yetersiz

2.4 MaxScript Tanımı

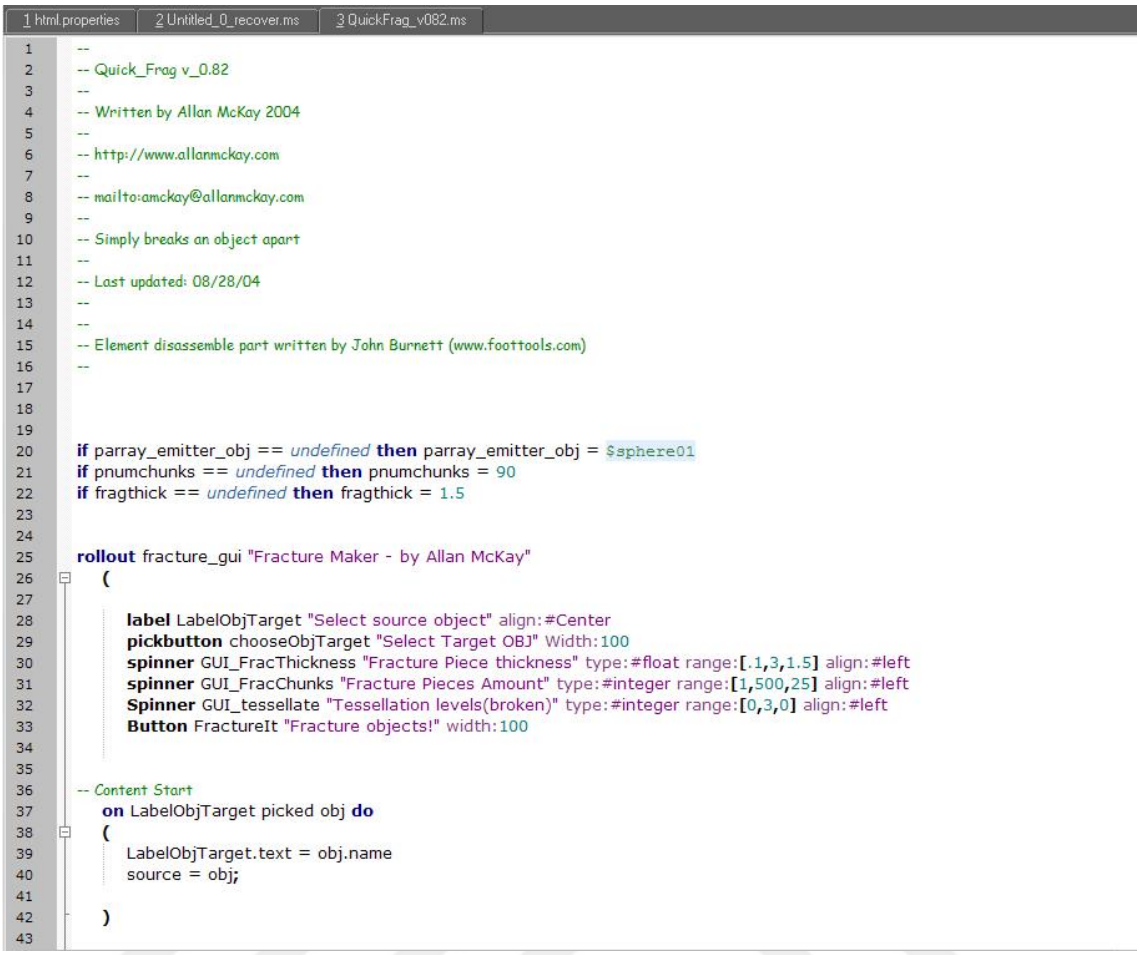
MaxScript ilk olarak 3D Studio MAX r2 içerisinde bir plug-in olarak ortaya çıktı. Daha sonradan 3DS Max'in çekirdek kodları içerine dahil edildi. Orijinal uygulama şu anki haline göre çok çok basit daha durumdaydı.

Fonksiyonellik ve kullanışlılık devamlı olarak her yeni sürümde geliştirildi. Önemli bir gelişme olarak Macroscript eklenmesi oldu. Bu, yerleşik kod olarak klavye kısa yolları, menüler ya da quad menüler aracılığı ile UI içerisinden çağrılan scriptlere olanak sağladı. Temel gelişme, Function Publising System (FPS) denilen sistemin kullanıcıların kendi

kodlarını MaxScript'te yaratmalarına olanak sağlaması oldu. Aslında MaxScript barındırdığı kod örnekleriyle, 3D matematik bölümleriyle ve genel ve özel programlama prensipleriyle beraber basit bir komut dilinden tam teşekküllü bir programlama diline gelmiştir. Ve en büyük gelişmelerin bir diğeri de MaxScript Community'nin yaptığı gelişmedir; sayısız internet sayfası, script veri tabanları, tartışma platformları (forum siteleri) ve diğer MaxScript ilişkili destek ve kaynaklar, birbirine sımsıkı bağlı olan profesyonel ağlardır.

MaxScript bir programlama dilidir. C++ dan daha basit ve daha kolaydır ama bütün programlama dilleri gibi mantık, gramer ve söz dizimleri gerektirir. MaxScript nesne tabanlı, olaya dayalı ve ifade-anlatım temellidir.

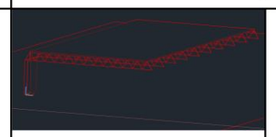
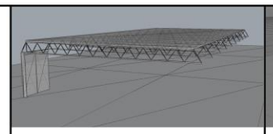
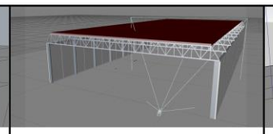
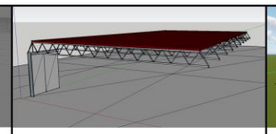
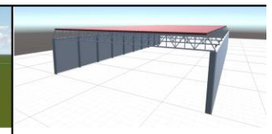
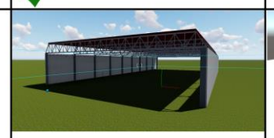
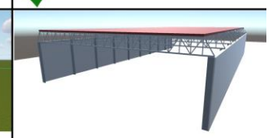
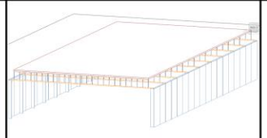
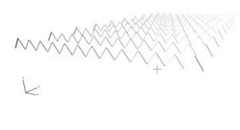
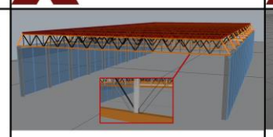
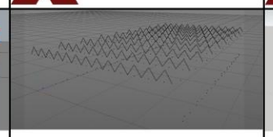
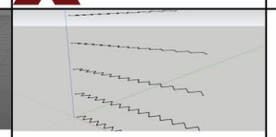
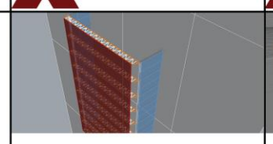
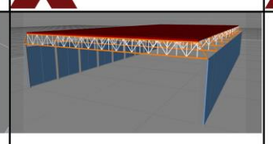
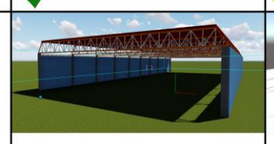
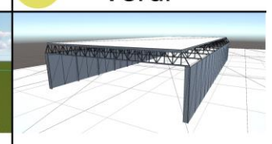
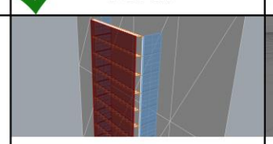
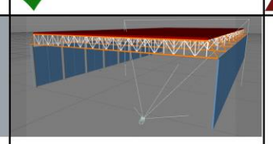
Script dil kendi programını kullanır. Basit anlamıyla MaxScript, 3DS Max'i kontrol eder. 3DS Max içerisinde yaptığınız hemen her şeyi MaxScript kullanarak yapabilirsiniz. Yani MaxScript, 3DS Max çalışma ekranında yapılabilen modelleme uygulamalarını, değiştirici uygulama işlemlerini, animasyon, malzeme yönetimi, sahnede kamera yönetimini, ışık etkilerini, render uygulamalarını, biped oluşturma ve canlandırmalarını, reactor uygulamalarını ve atmosfer etkilerini, yaratabilir ve kontrol edebilirsiniz. Hatta import/export objects and data, Max UI ayarları, flash ve activeX kullanımı ve .NET kontrollerinin kullanımı ileri seviye özellikleride MaxScript kullanarak rahatlıkla yapabiliriz. Tüm bunlar MaxScript kullanmak için iyi nedenlerdir. Bununla birlikte asıl güçlü yanı, işlemleri otomatikleştirmesidir. Kullanıcı, sık kullandığı işlemleri içeren bir kod yazarak, bu işlemleri tek tıklama ile gerçekleştirebilir [38].



Şekil 2.28 MaxScript ekran görüntüsü [39]

Bunların dışında 3DS Max çalışma ekranında yapılamayacak bazı karmaşık ve rastlantısal (random gibi) işlemleri başarı ile gerçekleştirmek mümkündür. Ayrıca 3DS Max çalışma ekranında kullanılan menü ve buton gibi araçları özelleştirme imkanı sunar. Yukarıda sayılan tüm özellikler için kullanıcı arayüzü oluşturulabilmektedir. Oluşturulan script dosyaları plug-in seklinde 3DS Max programına dahil edilebilir.

Script dosyaları ASCI ve binary dosya biçiminde başka programlara gönderilebilir ya da başka programlardan alınabilir. Sahnenin konumu ve durumu ile ilgili kontrol usulleri tanımlanabilir. Render işlemleri için toplu iş kodları tanımlanabilir. MaxScript komutlarını kullanarak belirli davranışlar kaydedilebilir [40].

	Autocad	Revit	Archicad	Rhino	Cinema4D	SketchUp	Lumion	Unity
3DS	 Yüzey Hataları Verdi	Format Desteklenmiyor	Format Desteklenmiyor	 Yüzey Hataları Verdi	 Hatasız	 Yüzey Hataları Verdi	 Hatasız	 Hatasız
DAE	Format Desteklenmiyor	Format Desteklenmiyor	Format Desteklenmiyor	Format Desteklenmiyor	Format Desteklenmiyor	Format Desteklenmiyor	 Hatasız	 Hatasız
DWG	Format Desteklenmiyor	 Obje Hatası Verdi	 Obje Hatası Verdi	 Basit Yüzey Hataları Verdi	 Obje Hatası Verdi	 Obje Hatası Verdi	Format Desteklenmiyor	Format Desteklenmiyor
FBX	Format Desteklenmiyor	Format Desteklenmiyor	Format Desteklenmiyor	Format Desteklenmiyor	Format Desteklenmiyor	Format Desteklenmiyor	 Hatasız	 Arayüz Hatası Verdi
OBJ	Format Desteklenmiyor	Format Desteklenmiyor	Format Desteklenmiyor	 Eksen Hatası Verdi	 Hatasız	Format Desteklenmiyor	 Hatasız	 Hatasız
WRL	Format Desteklenmiyor	Format Desteklenmiyor	Format Desteklenmiyor	 Eksen Hatası Verdi	 Hatasız	Format Desteklenmiyor	Format Desteklenmiyor	Format Desteklenmiyor

Çizelge 2.2 Export edilen modelin incelenen programlarda hangi formatların desteklendiğini gösteren çizelge

BÖLÜM 3

PARAMETRİKLEŞTİRİLMİŞ STRÜKTÜR SİSTEMLERİ (PSS)

3.1 GAMS ve MaxScript İlişkisinin Kurulması

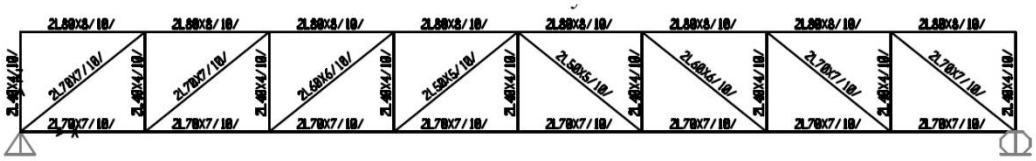
Geniş açıklıklı mekan sistemleri, matematiksel anlamıyla, tüm bileşenleriyle birbirine bağlı, değişen bir bileşenin diğerini etkilediği parametrik sistemlerdir. Değişen sistem tipi ya da geçilen mesafe gibi değişkenler kolon ebatlarından kiriş boyutlarına, profil boyutlarından kullanılan profil tipine kadar pek çok bileşeni etkilemekte ve birbirlerine referans oluşturmaktadır. Dolayısı ile tüm bu etkenler birbirlerine matematiksel değerlerle bağlıdır.

Düz kiriş, kemer kiriş, kafes kiriş ve uzay kafes kiriş olmak üzere dört ana başlık altında topladığım program arayüzünde, kafes kiriş sistemlerini script diline aktardım. Kullanılabilecek çeşitli profil tipleri de olduğu ve en net görseli elde edebilmek adına kutu formunda oluşturabilmek için L profil tercih edildi. Sonuç görselin de gerçeğine yakın olabilmesi için gerçek veriler kullanıldı.

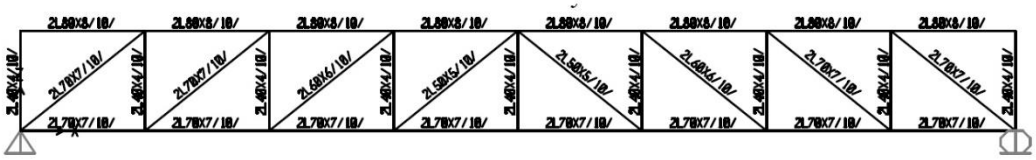
Kafes kirişlerin orta yükseklikleri, ekonomik boyutlandırma açısından en uygun aralık olan L/12-L/16 aralığında olacak şekilde hesaplanmıştır [51].

Howe Tipi Kafes Kiriş Çözümleri

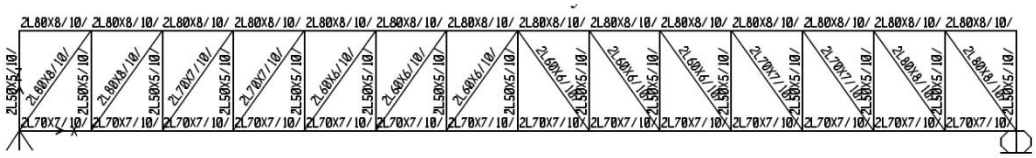
Paralel Başlık



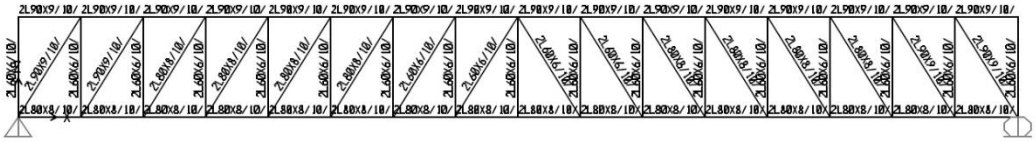
Kafes kiriş çubuklarının kesitleri
Howe tipi kafes sistemin SAP2000 çözümü (L = 15 m)



Kafes kiriş çubuklarının kesitleri
Howe tipi kafes sistemin SAP2000 çözümü (L = 20 m)



Kafes kiriş çubuklarının kesitleri
Howe tipi kafes sistemin SAP2000 çözümü (L = 25 m)

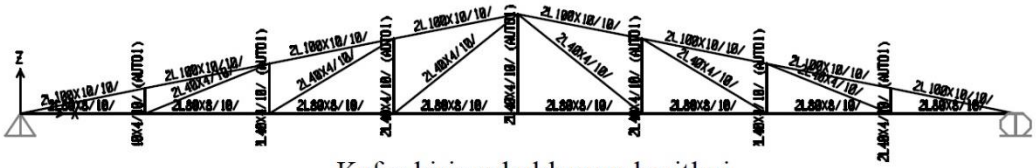


Kafes kiriş çubuklarının kesitleri
Howe tipi kafes sistemin SAP2000 çözümü (L = 30 m)

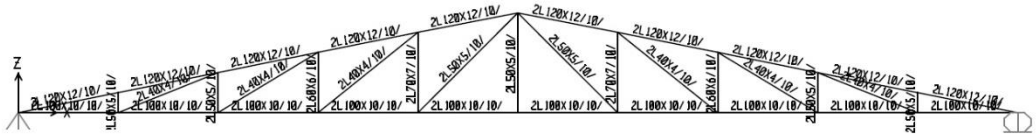
Şekil 3.1 Howe Tipi paralel başlık kafes sistem SAP2000 çözümü [52]

Howe Tipi Kafes Kiriş Çözümleri

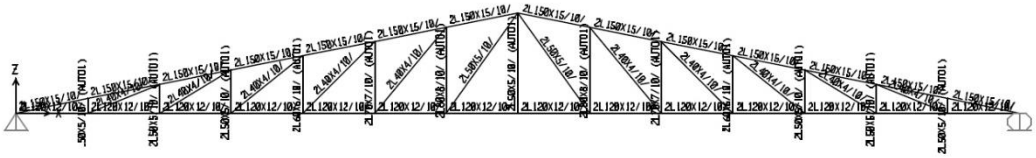
Üçgen Başlık



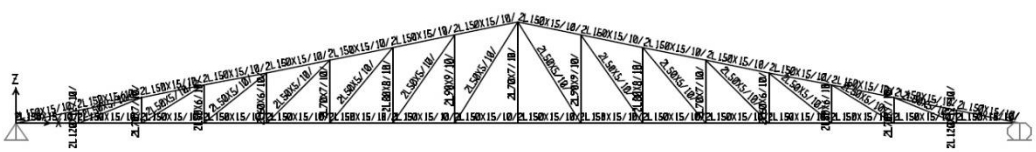
Kafes kiriş çubuklarının kesitleri
Howe tipi kafes sistemin SAP2000 çözümü (L = 15 m)



Kafes kiriş çubuklarının kesitleri
Howe tipi kafes sistemin SAP2000 çözümü (L = 20 m)



Kafes kiriş çubuklarının kesitleri
Howe tipi kafes sistemin SAP2000 çözümü (L = 25 m)



Kafes kiriş çubuklarının kesitleri
Howe tipi kafes sistemin SAP2000 çözümü (L = 30 m)

Şekil 3. 2 Howe tipi üçgen başlık kafes sistem SAP2000 çözümü [52]

Diğer başlık tiplerinin SAP2000 çözümleri Ekler bölümünde verilmiştir.

Sıradan modelleme tekniği ile yapının modellemesi sırasında kullanılan tüm bileşenler, tek tek modellenerek birleştirilir. Bu işlem, yapının karmaşıklığı ve büyüklüğü ile orantılı olarak zaman almaktadır. Tüm bu işlemlerden sonra yapılan bir değişiklikler ya da olabilecek hatalar, sistemin neredeyse en baştan modellenmesine neden olabilir. Bu da kullanıcıya gereksiz zaman ve emek kaybı olarak geri döner.

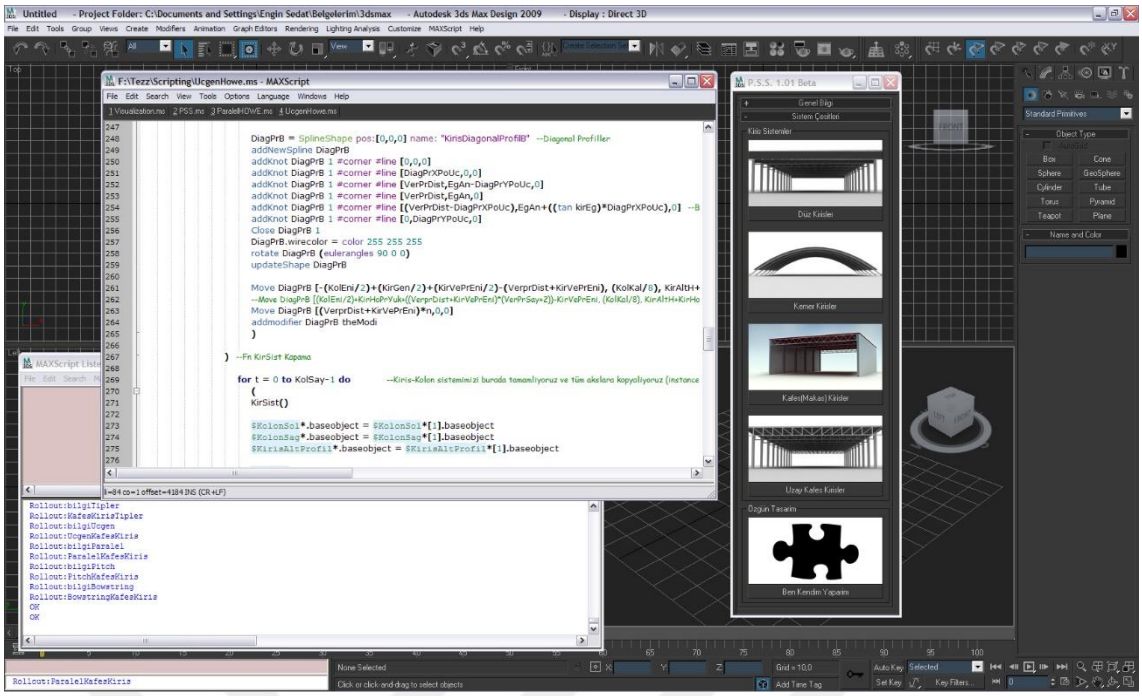
Parametrik alt yapıya sahip yapılarda, scripting aslında kullanıcıya büyük kolaylık sağlar. Yukarıda belirtilen matematiksel veriler yalnızca bir kez kodlama mantığı içerisine formül edilmesi yeterlidir. İstediğiniz kadar deneme yanılma yapabilir, çeşitlik sağlayabilmektedir. Projenizin son halinin modeli, 3d fotorealistik görselleri ve genel sistem bilgileri elinizde olabilir.

Bu konuda bize yardımcı olabilecek olan MaxScript kodlama sistemi, kontrolünün tamamiyle sizin elinizde olduğu, yapılacak işlerin sizin hayal gücünüzle sınırlı olduğu ve kendinize ait bir arayüz oluşturabileceğiniz program eklentisidir. Sonuç ürüne ulaşabilmek için uzun zaman harcamak yerine size sadece birkaç tıklama ile sonuca ulaşabilmeniz sağlanmıştır.

3.2 PSS Programı

PSS programı, MaxScript kod dizileriyle oluşturulmuştur. Sadece 3DS Max programı altında çalışmaktadır. 2015 ve 2016 versiyonlarında hatasız bir şekilde çalışmaktadır. Daha eski vesiyonlarda ufak tefek uyumsuzluklar görülebilir.

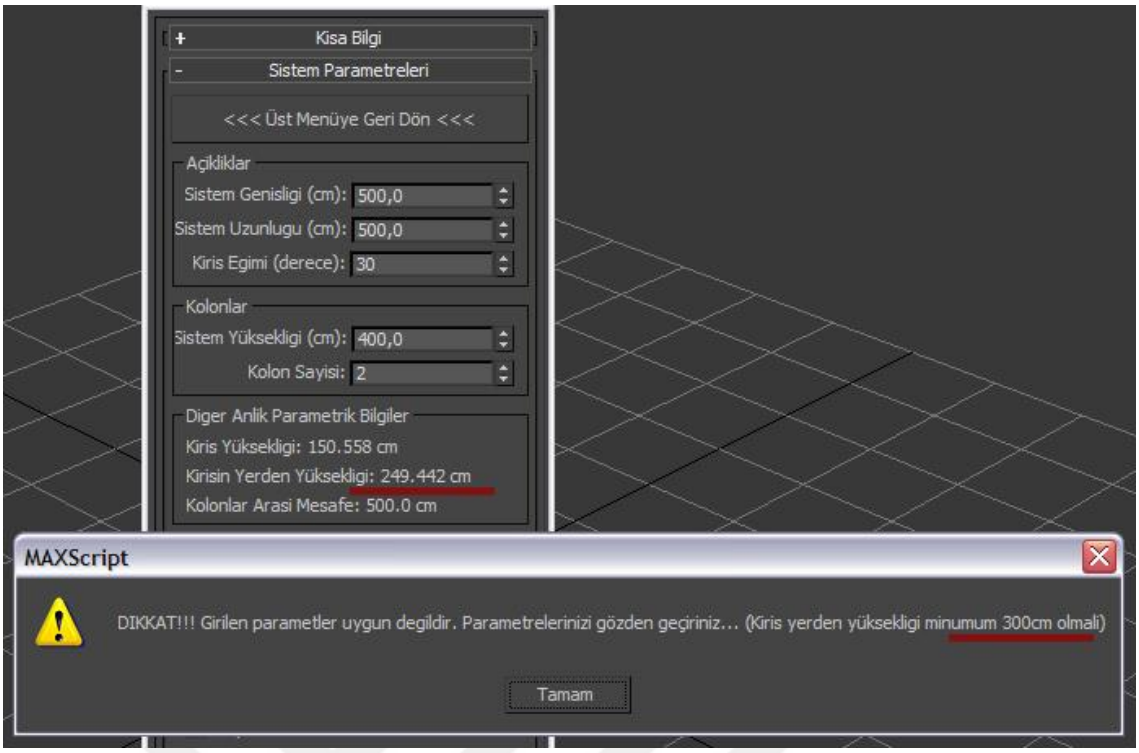
Oluşturulan program ya da bir diğer deyişle programcık, MaxScript kod dizileri mantığında işlenen matematematiksel verilerin, parametrik alt yapıya sahip geniş açıklıklı mekan sistemlerini tanımlamasıyla oluşturulmuştur. Belli geometrik formların ve temel statik hesaplarının programa girişi için belirli tanımlamalar oluşturulmuş ve sonrasında MaxScript ile kodlanmaya başlanmıştır.



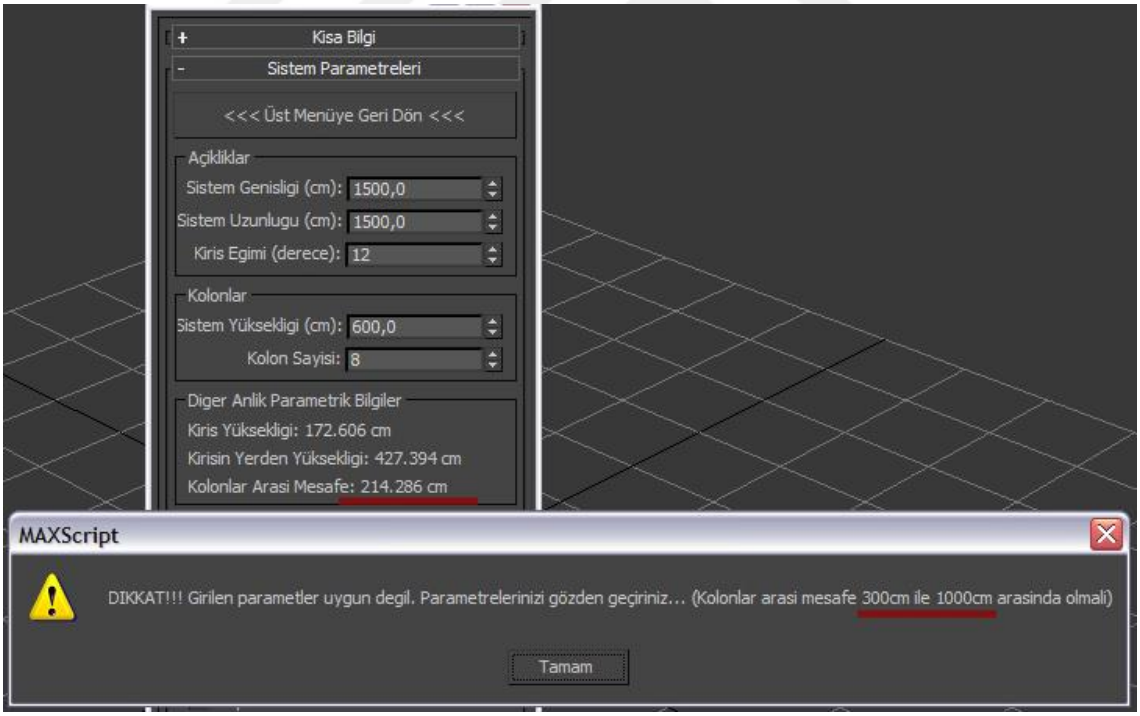
Şekil 3.3 P.S.S. Programı arayüz ve kod dizisi ekran görüntüsü

Programı kullanmak için ciddi bir 3DS Max bilgisine ihtiyaç duyulmayacak şekilde tasarlanmıştır. Kullanıcı oluşturmak istediği sistemi görsel ve yazılı yönlendirmelerle kolaylıkla oluşturabilir.

Kullanıcı oluşturmak istediği sistem tipini seçtikten sonra, program kullanıcıdan sistem uzunluğu, geçilmesi istenen açıklık, aks sayısı, yapı yüksekliği ve varsa çatı eğimi gibi verileri ister. Girilen bu veriler dahilinde değişecek olan, kiriş yüksekliği, kirişin yerden yüksekliği ve kolonlar arası mesafe gibi tasarımı etkileyen ölçüleri henüz model oluşmadan önce aktif olarak görebilmektedir. Eğer kullanıcının girdiği değerler sonucunda oluşacak yapı kullanıma uygun olmayan ölçülerde oluşuyor ise; program modeli oluşturmaz ve kullanıcıyı uyararak hatasını düzeltmesini ister.



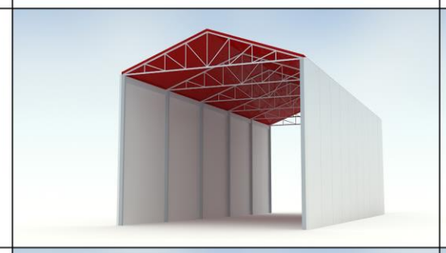
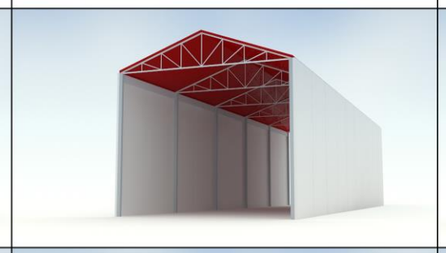
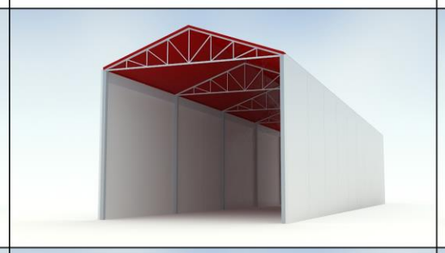
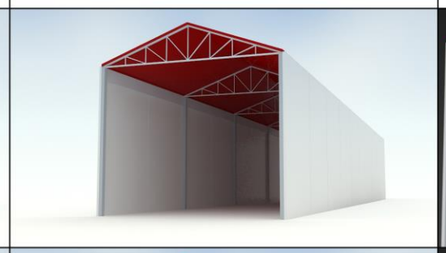
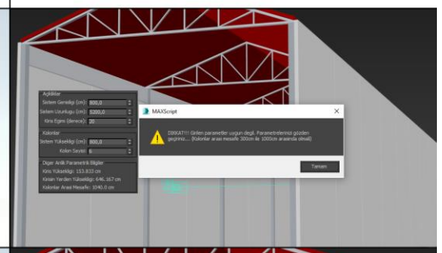
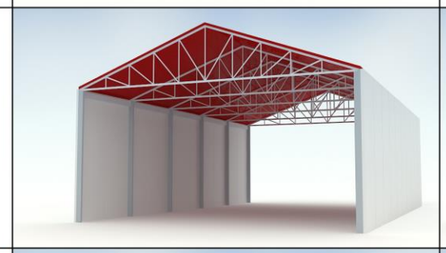
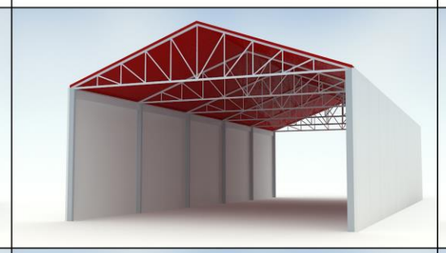
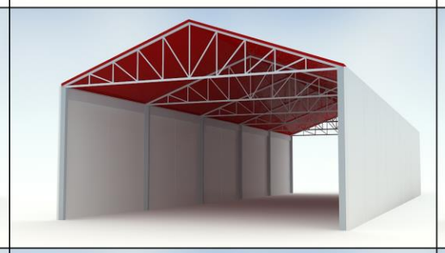
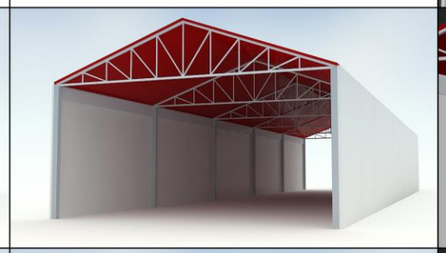
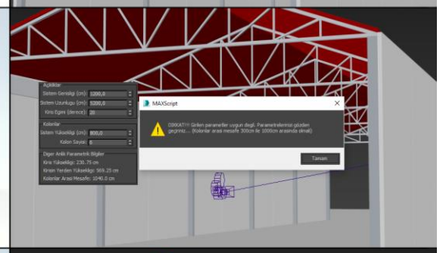
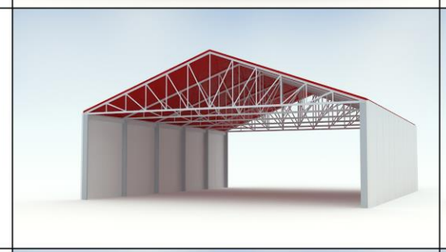
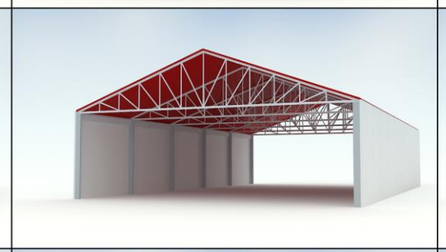
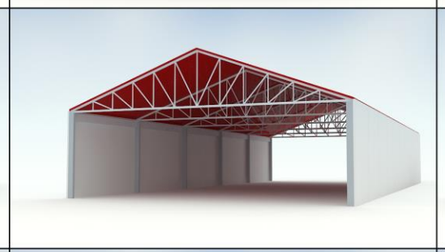
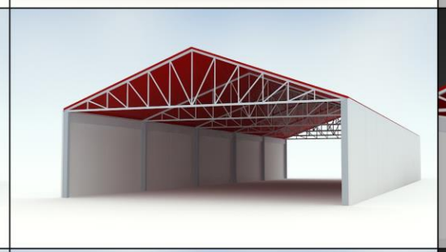
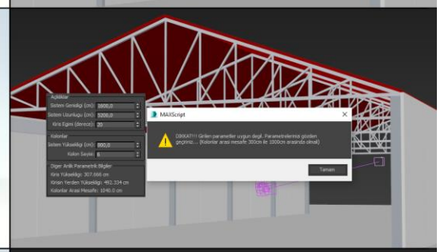
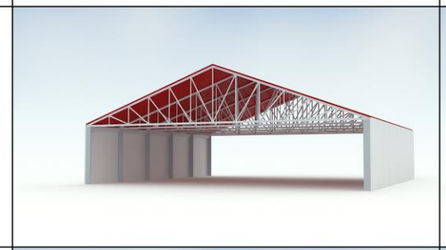
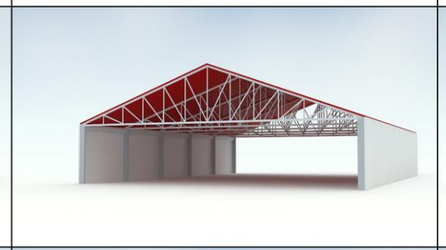
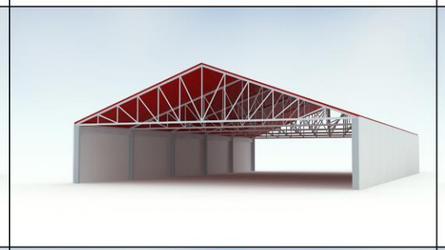
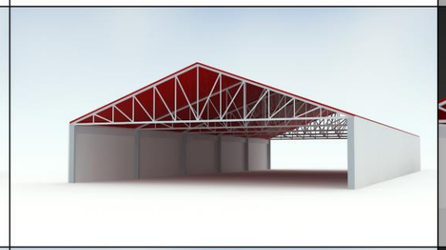
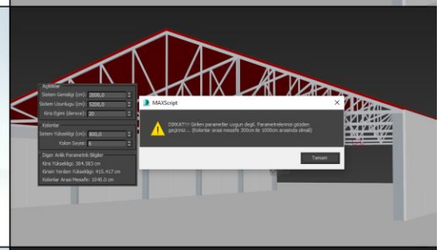
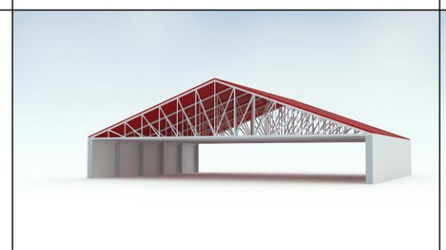
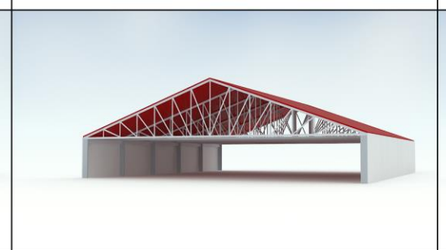
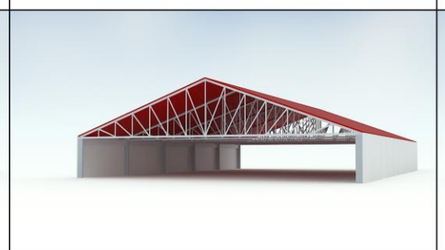
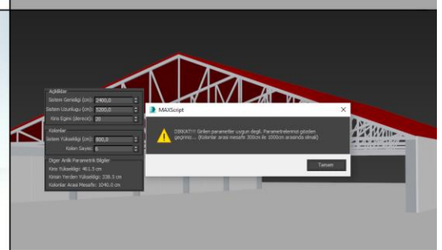
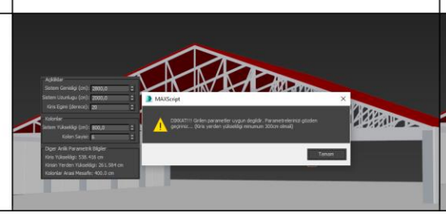
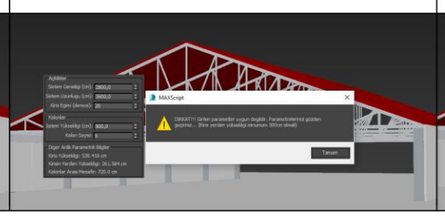
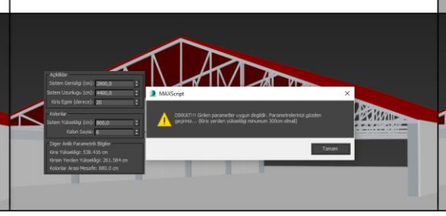
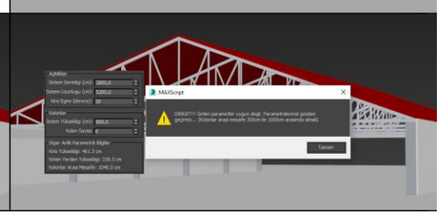
Şekil 3. 4 P.S.S. Programı hata ekranı – kiriş yüksekliği yetersiz olduğunda



Şekil 3. 5 P.S.S. Programı hata ekranı – kolonlar arası mesafe çok az veya çok fazla olduğunda

PSS kullanımı kolaylaştırmak için kendi arayüzüne sahiptir. Programın temel amacı en hızlı şekilde modeli oluştumak ve basit bir sunum olanağı sağlamak olduğu için görsel

seçenekler de eklenmiştir. Sadece taşıyıcı sistemi görmek istenildiği takdirde cephe ve çatı elemanlarını kaldırılabilir. İç mekan ve dış mekan olmak üzere 2 adet kamera sahneye yerleştirilir ve bakış açısı kullanıcı tercihine bırakılır. Pek çok 3ds Max kullanıcısı V-ray render motorunu kullanmaktadır fakat her kullanıcı V-ray plug-inine sahip değildir. Bu sebeple de default(standart) render seçeneği de arayüze eklenmiştir. Render işlemi standart ya da v-ray olarak seçilebilmektedir ve malzemeler de ona göre atanmaktadır. Kullanıcı ışık render ayarları ile uğraşmaz; render al seçimini gerçekleştirmesi yeterlidir. Oluşan model 3ds Max harici bir programda kullanılacaksa eğer, export seçeneklerinden tercih edilen formatın seçilmesi yeterlidir. En çok tercih edilen yedi farklı uzantı olanağı sağlanmıştır. (".3ds", ".Fbx", ".Dwg", ".Dxf", ".Obj", ".Dae", ".Wrl"). Kullanıcının, export modelinin kaydedildiği klasöre hızlı bir şekilde erişebilmesi için de buton eklenmiştir.

	20 metre	28 metre	36 metre	44 metre	52 metre
8 metre					
12 metre					
16 metre					
20 metre					
24 metre					
28 metre					

Çizelge 3.1 Geçilen açıklık ve sistem uzunluğu değişkenleri sonuç çizelgesi (6 aks üzerinden oluşturulmuştur.)

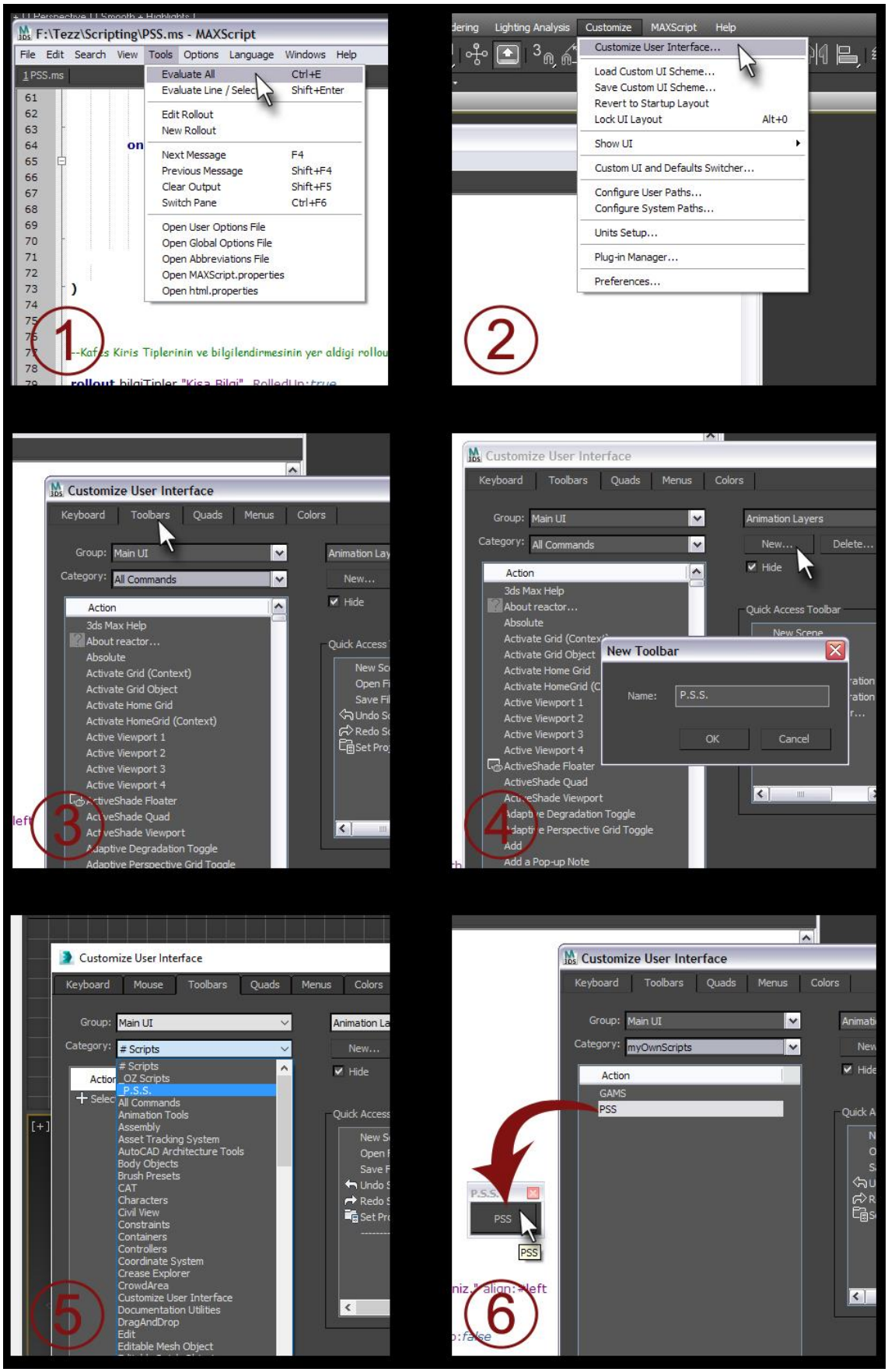
3.3 PSS Programı Kullanımı

Henüz geliştirme aşamasında olduğu için, program, bir plug-in gibi “install” işlemi ile değil, 3ds Max içerisinde yapacağımız birkaç işlem ile kurulumu yapılacak şekilde hazırlanmıştır.

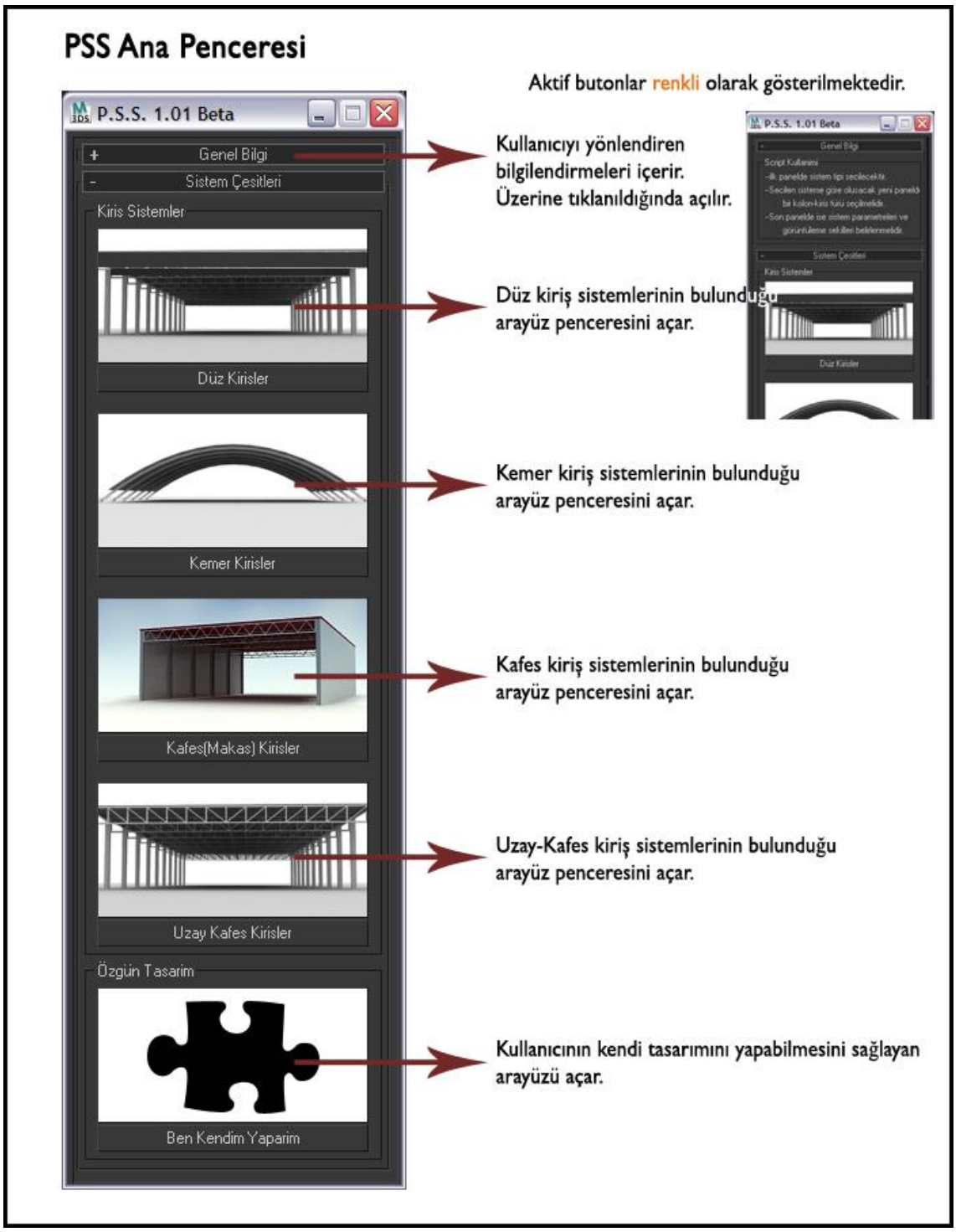
Aşağıdaki şekilde de (Şekil 3.6) görebileceğiniz 6 aşama var. Bu aşamalar sırasıyla;

- MaxScript editör penceresinde PSS.ms dosyasını açalım ve kendi penceresindeki “Tools” menüsünden “Evaluate All” seçimini yapalım.
- 3ds Max programında “Customize” menüsünden “Customize User Interface” seçimini yapalım.
- Karşımıza çıkan pencereden “Toolbars” sekmesini seçelim.
- “New” butonuna tıklayalım ve istediğimiz ismi vererek kendi araç çubuğumuzu (toolbar) oluşturalım.
- “Category” isimli düşer menü içinden “_P.S.S.” i seçelim.
- Son olarak, “_P.S.S.” i seçtikten sonra karşımıza çıkan “P.S.S.” metnini sürükleyerek oluşturduğumuz araç çubuğumuza bırakıyoruz.

şeklinde tanımlanmıştır. Artık kullanıcı oluşturduğu araç çubuğunu istediği yere sürükleyip bırakabilir ve sonraki oturumlarında da bıraktığı yerden erişim sağlayabilir.

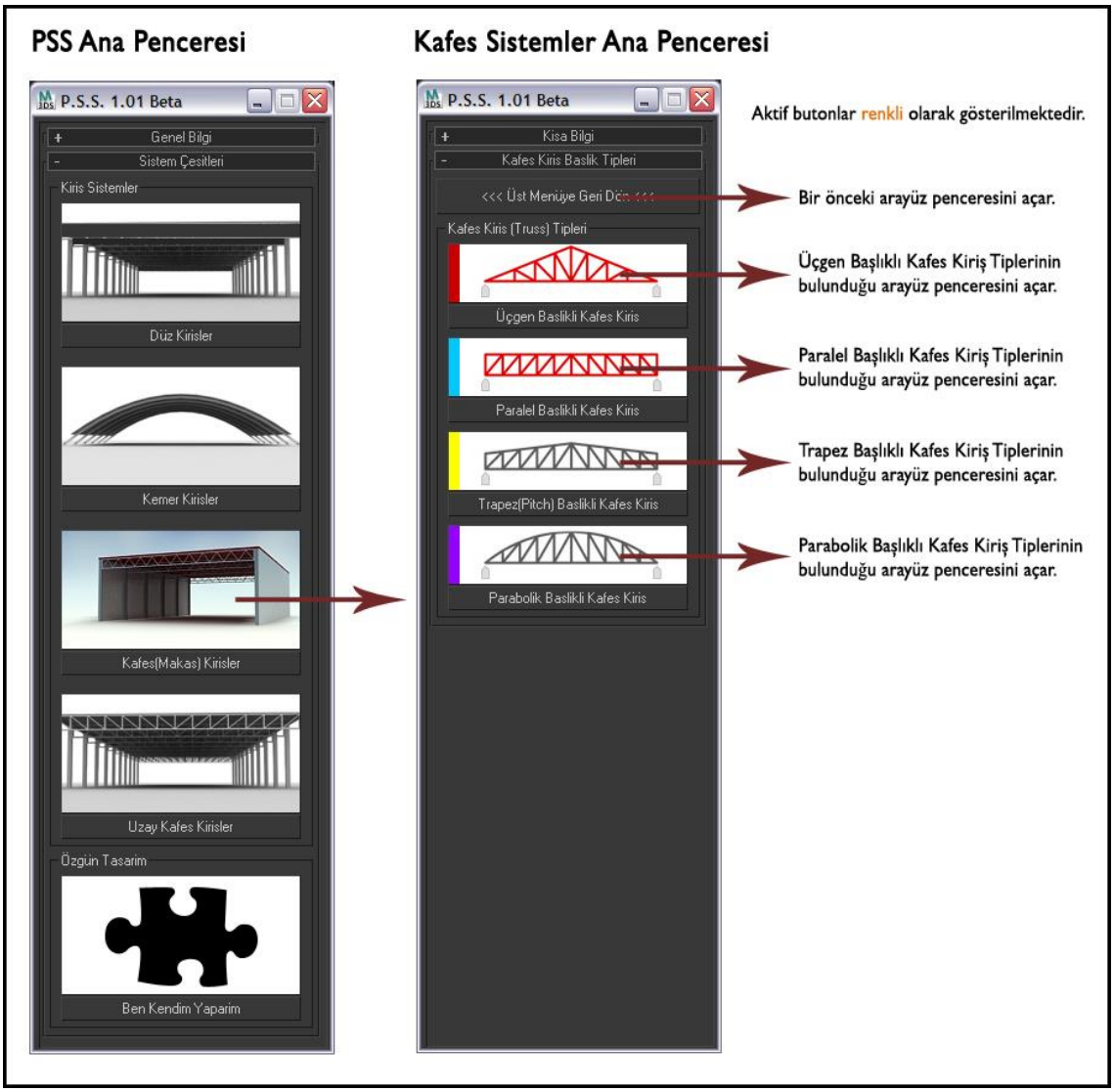


Şekil 3.6 P.S.S. Programı kurulumunu anlatan ekran görüntüleri



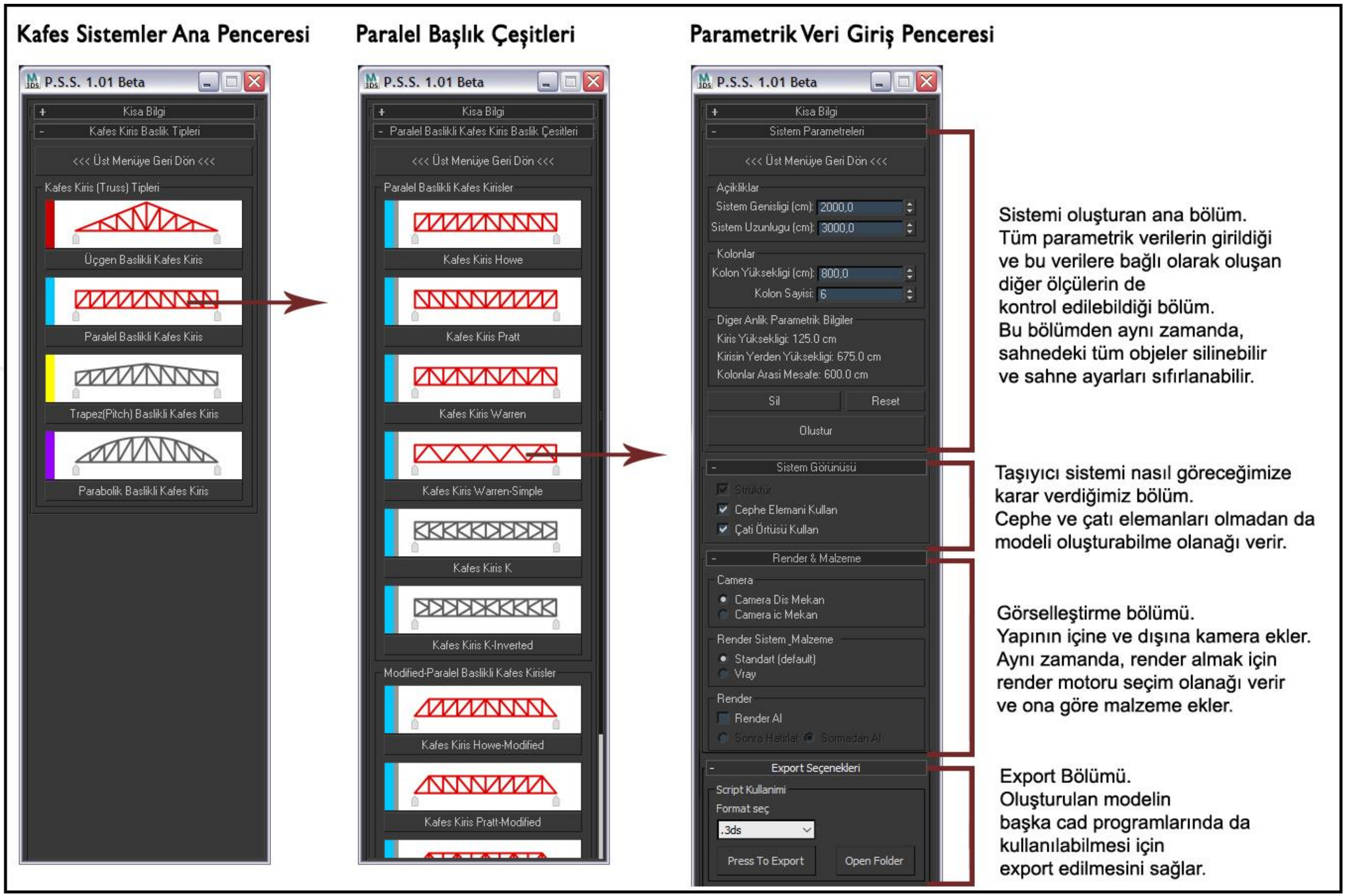
Şekil 3.7 P.S.S. Programı arayüzü ve açıklaması

Kullanıcı uygulamayı açtığında ilk olarak yukarıdaki arayüz ile karşılaşır. Görsel ağırlıklı ve anlaşılması kolay olan bu arayüzde, kullanıcıdan tasarımında tercih ettiği taşıyıcı sistem tipini seçmesi beklenmiştir.



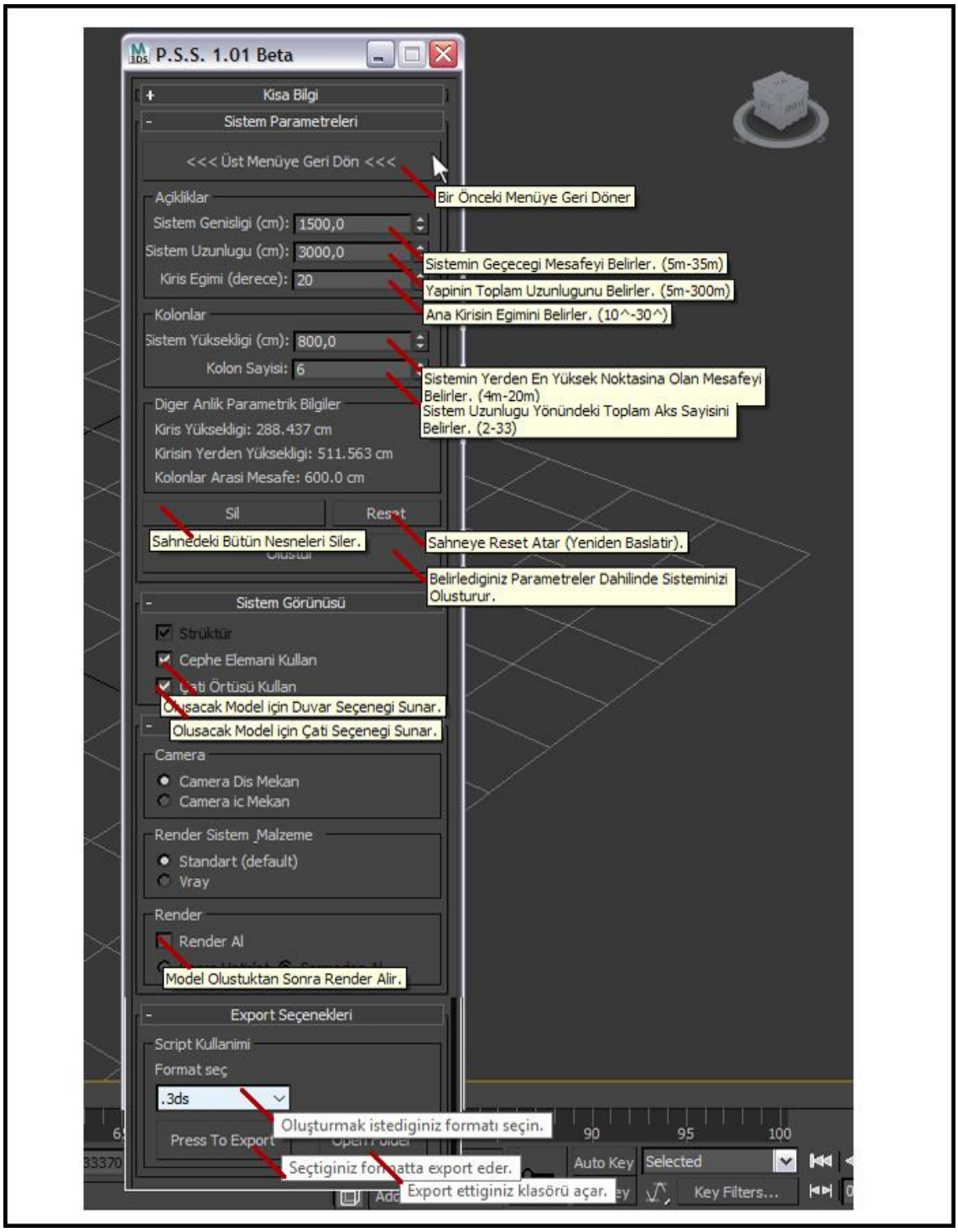
Şekil 3.8 P.S.S. Programı arayüzü – kafes sistemler menüsüne geçiş ve açıklaması

Bu aşamada sistemlerden aktif olan “Kafes(Makas) Kirişler” seçildikten sonra 4 ana başlıkta toplanmış olan Kafes Kiriş Tipleri görülmektedir. Bunlardan en çok tercih edilen üçgen ve paralel başlıklı kafes kirişler için yazılım yapılmıştır. Burada yapılan yazılım, şu an aktif olmayan diğer iki başlık için de kodlanabileceği gibi, aynı zamanda, burada yer almayan başlık tipleri içinde uygulanabilecek şekilde hazırlanmıştır.



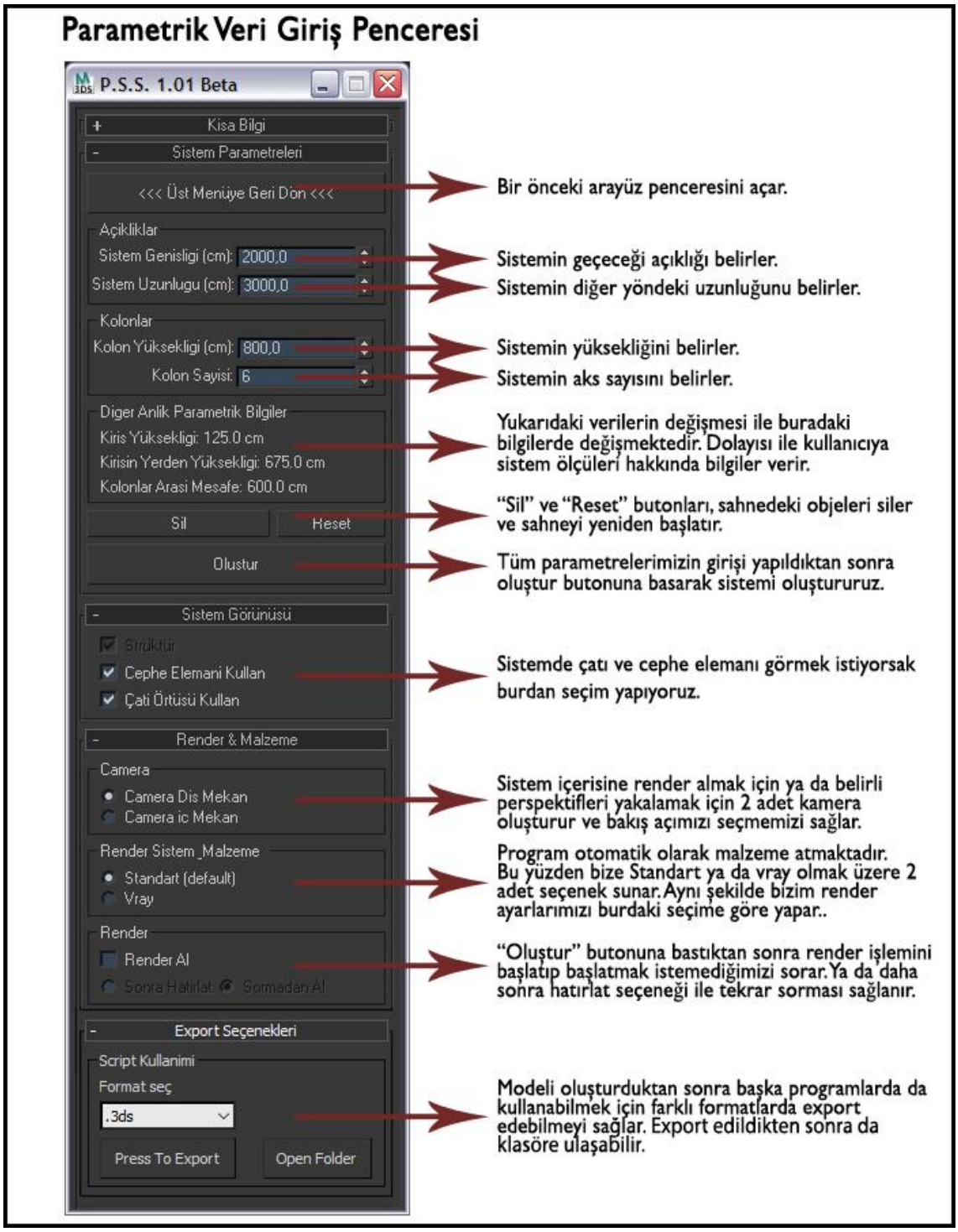
Şekil 3.9 P.S.S. Programı arayüzü – paralel başlık çeşitleri ve arayüz bölüm açıklamaları

Paralel başlıklı kiriş sistemini seçen kullanıcı en çok kullanılan 10 farklı kiriş tipinin 7sini modelleyebilmektedir. Bu tabloda kullanıcının karşılaştığı arayüzün hangi bölümünde nereyi kontrol ettiği anlatılmıştır.



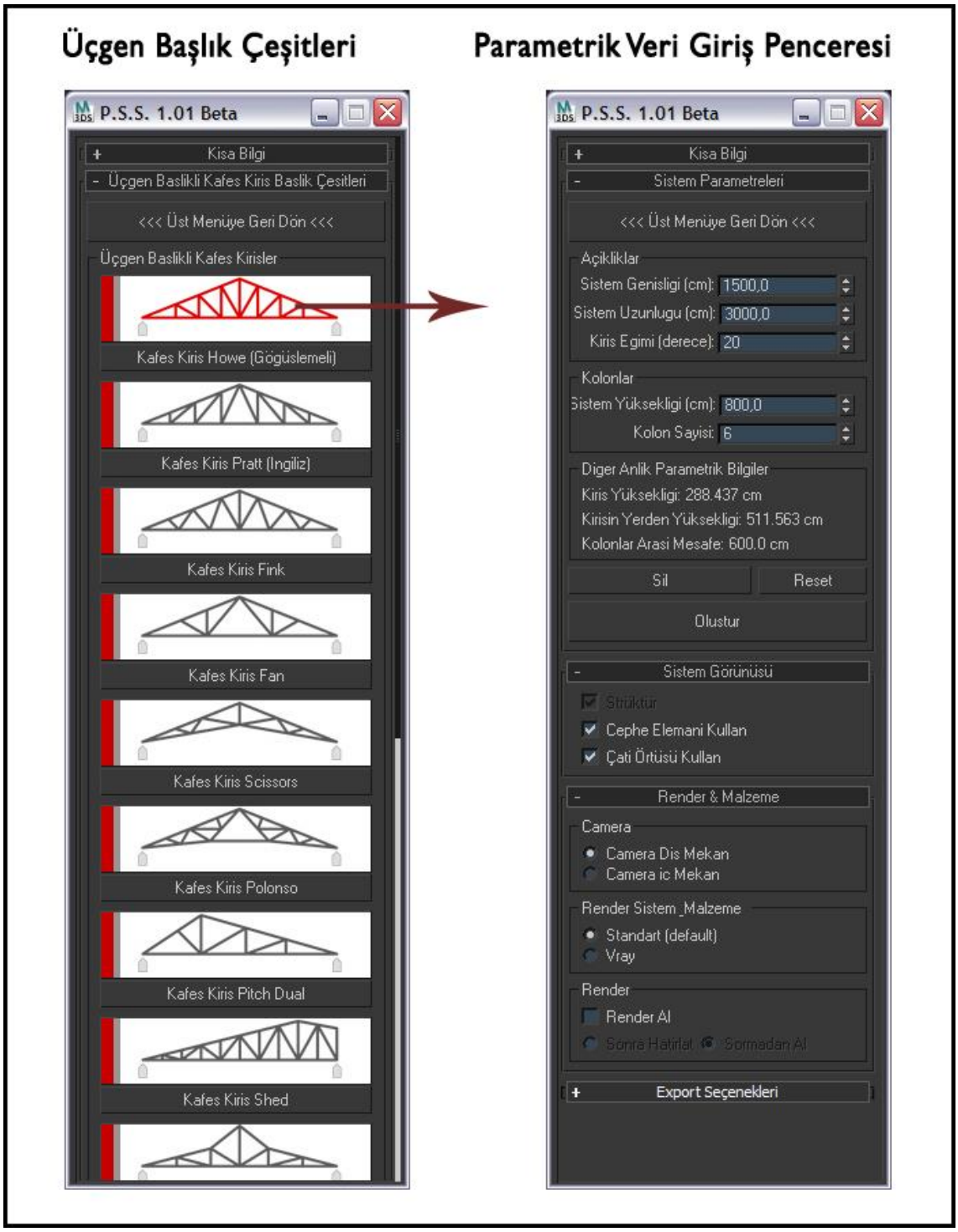
Şekil 3.10 P.S.S. Programı arayüzü – yardımcı pop-upların toplu görüntüsü

Kullanıcı için tanımlanmış küçük baloncukların hepsini bir arada görebilmek için hazırlanmıştır. İmleci her hangi bir komutun üzerinde bir müddet bekletmesi yeterlidir. Kullanıcı hangi komutun ne işe yaradığını bilemediği durumlarda kolaylık sağlaması için oluşturulmuştur.



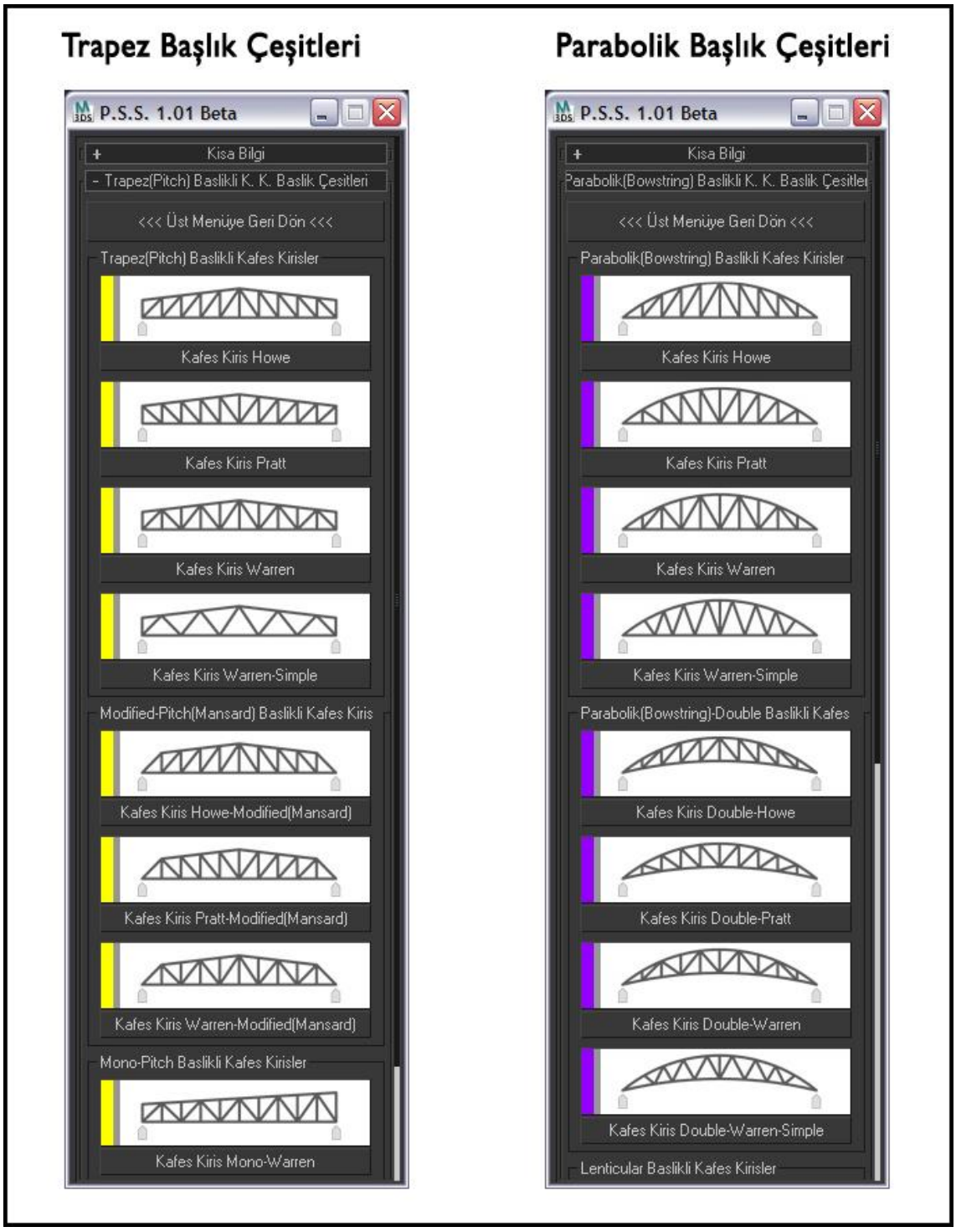
Şekil 3.11 P.S.S. Programı arayüzü – parametrik veri penceresi açıklamaları

Bu görselde, kullanıcının modeli oluşturmadan önce karşılaştığı pencerede yer alan tüm kontrollerin açıklayıcı bilgilerine yer verilmiştir.



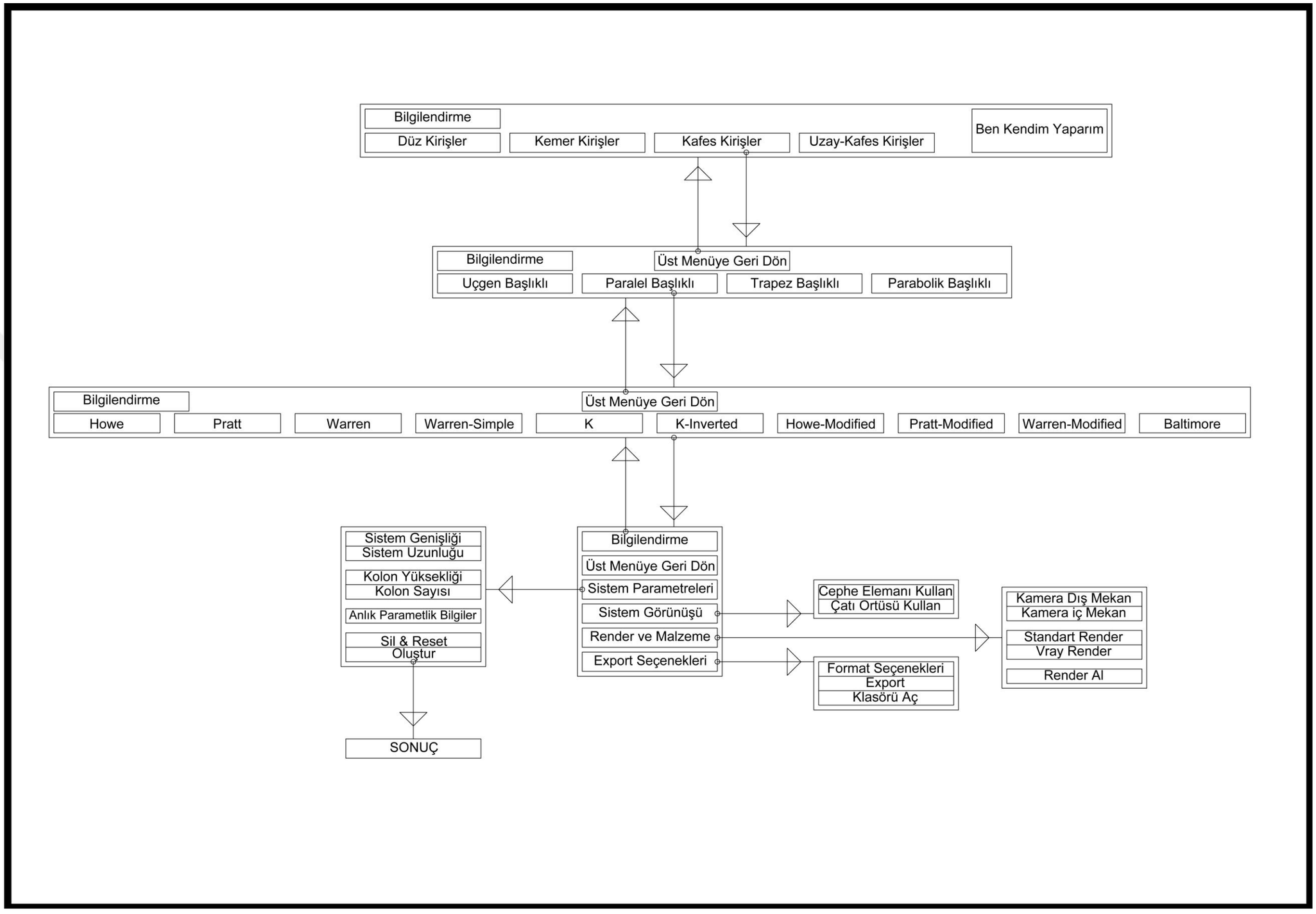
Şekil 3.12 P.S.S. Programı arayüzü – üçgen başlıklı-howe penceresine geçiş

Üçgen başlıklı kafes sistemlerinden aktif olan Howe tipi kafes kiriş ve sonrasında karşımıza çıkan arayüz. Paralel kiriş sistemlerinden farklı olarak kiriş eğimini de belirtebileceğimiz alan mevcuttur.

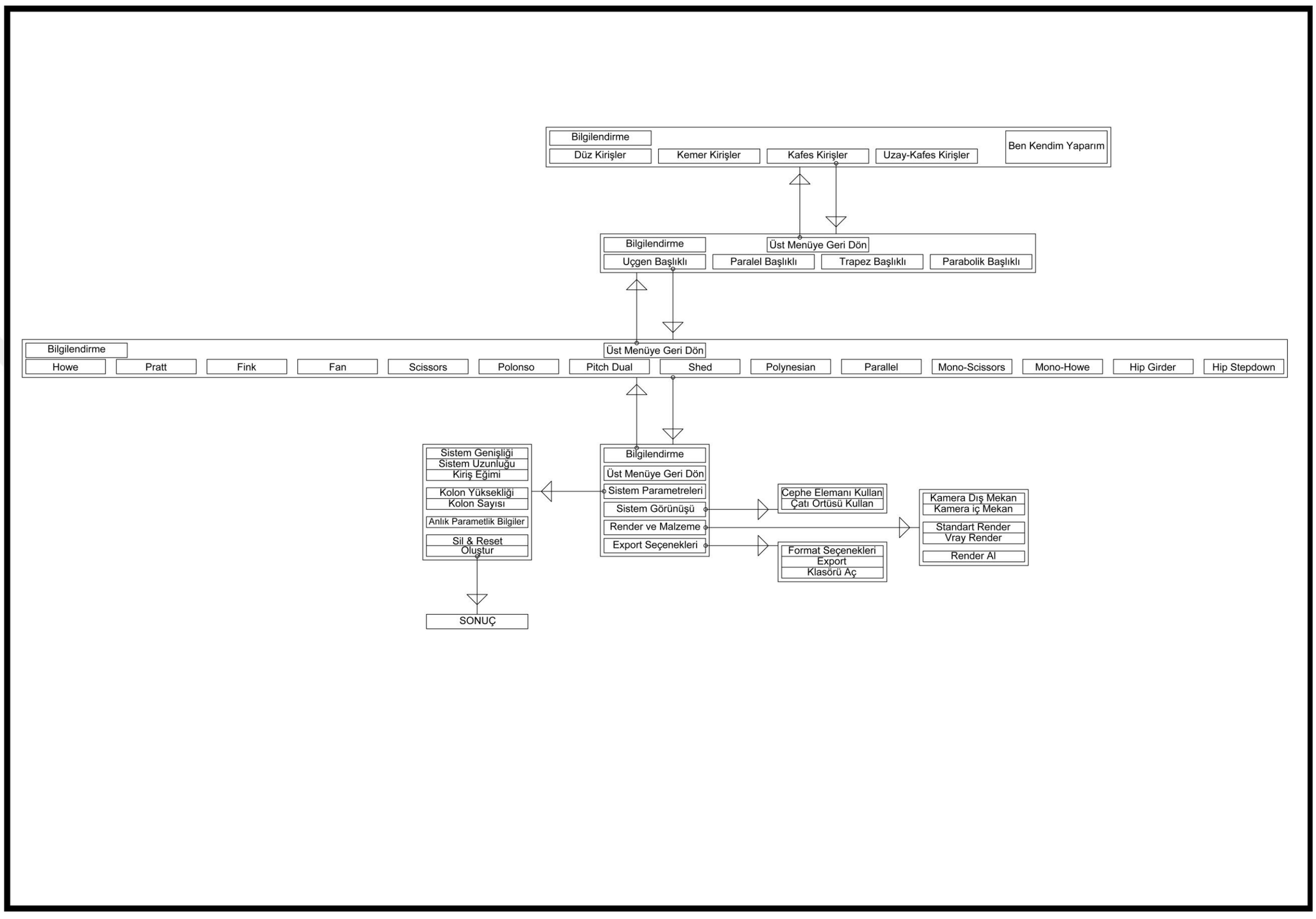


Şekil 3.13 P.S.S. Programı arayüzü – trapez ve parabolik başlık çeşitleri

Bu bölümde ise kullanıcının modelini oluşturabileceği fakat aktif olarak kullanılabilecek durum olmayan kafes sistemler gösterilmiştir.

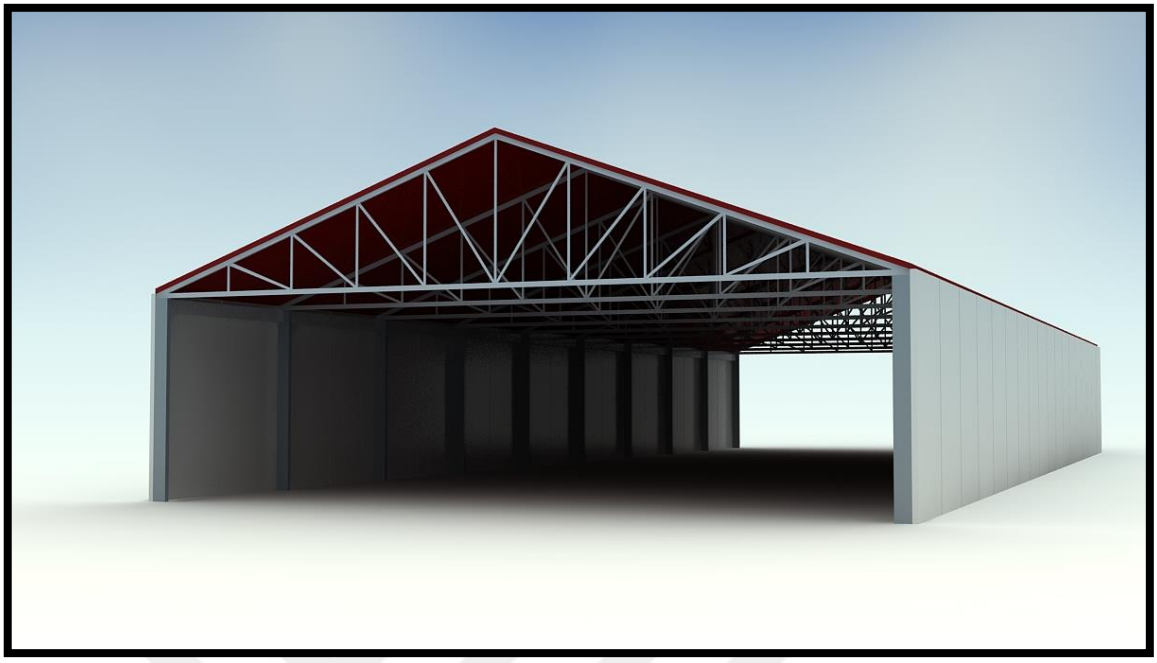


Şekil 3.14 P.S.S. Programı paralel başlıklı kafes kiriş sistemler akış diyagramı



Şekil 3.15 P.S.S. Programı üçgen başlıklı kafes kiriş sistemler akış diyagramı

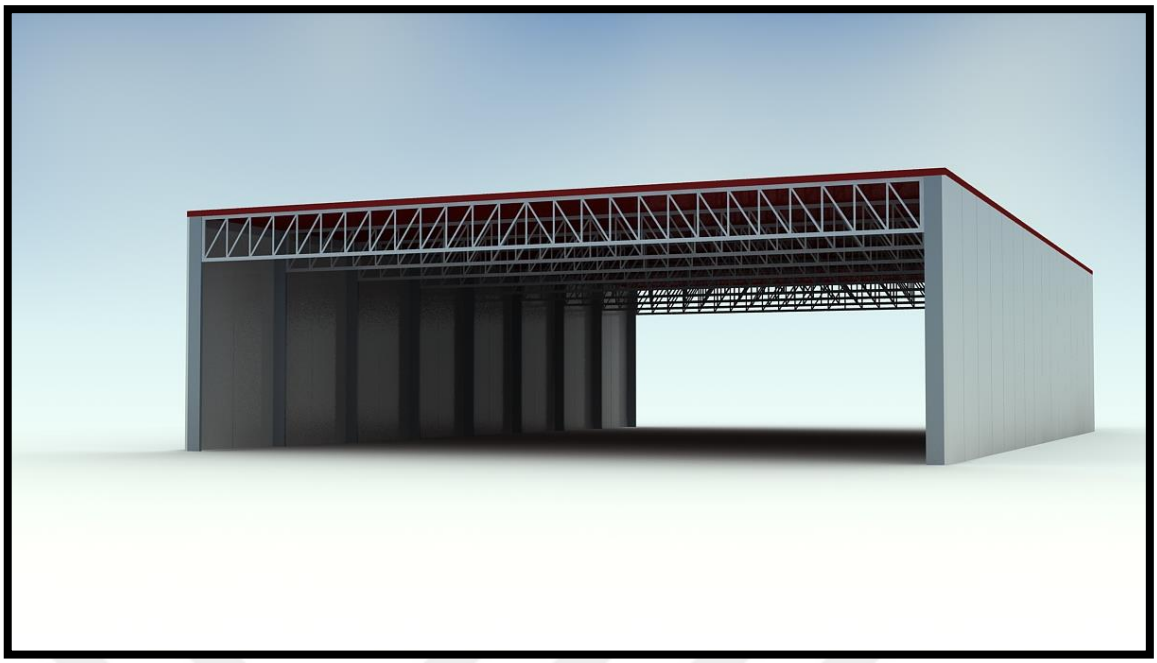
3.4 PSS Programı ile Yapılmış Örnekler



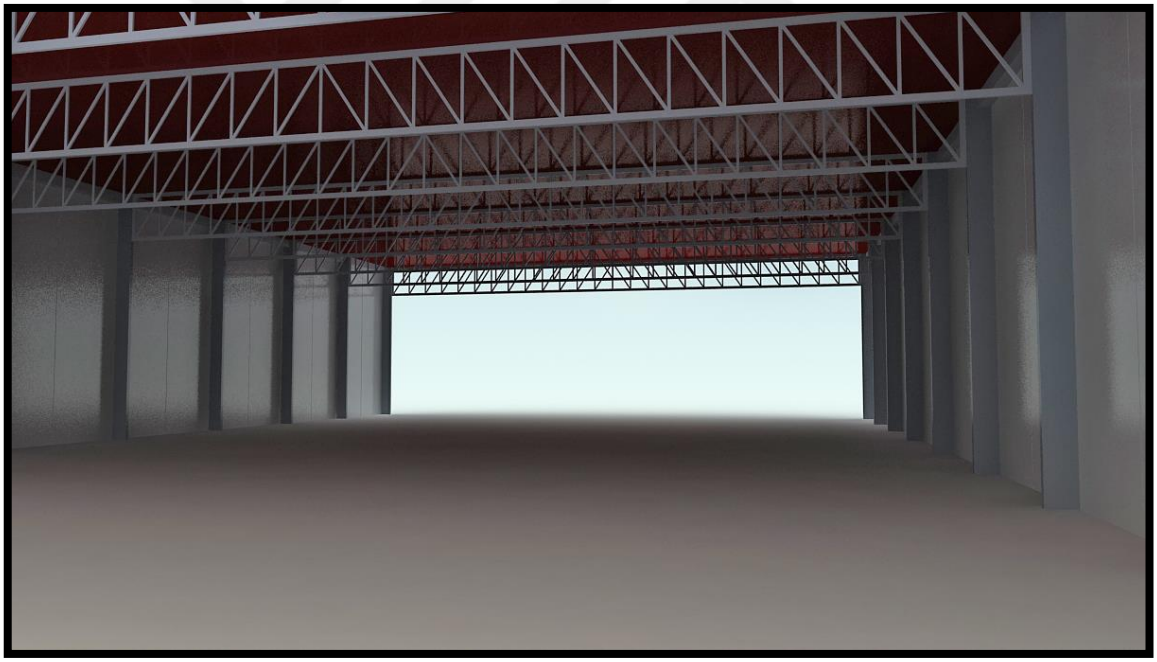
Şekil 3.16 Üçgen başlıklı howe kafes kiriş görüntüsü – dış perspektif



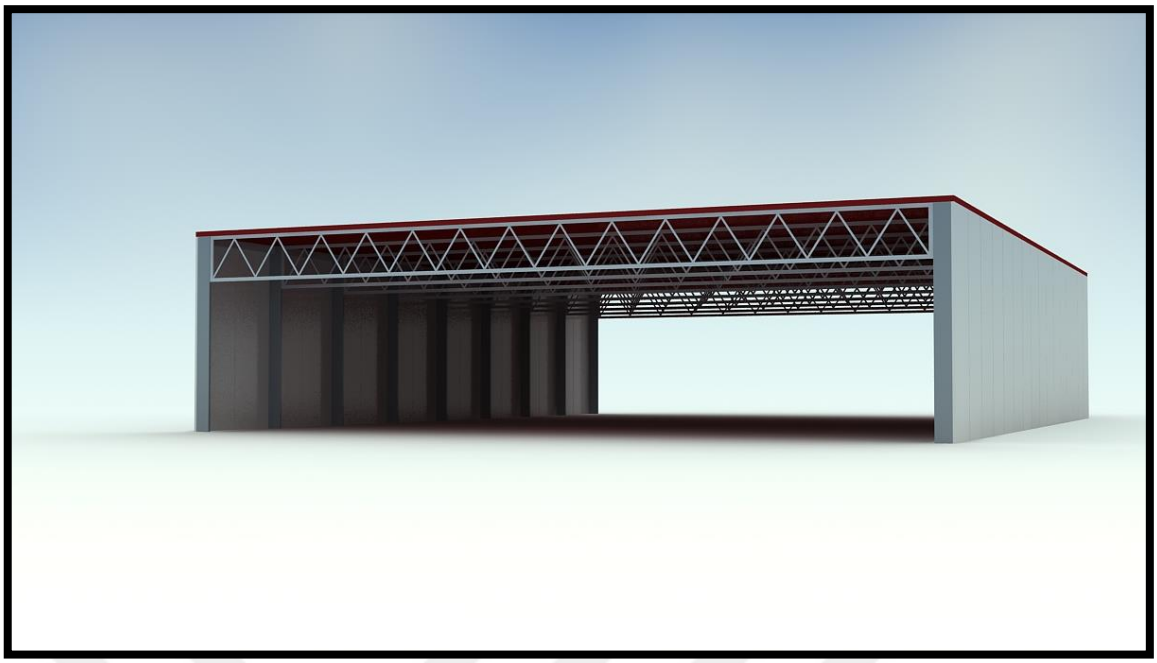
Şekil 3.17 Üçgen başlıklı howe kafes kiriş görüntüsü – iç perspektif



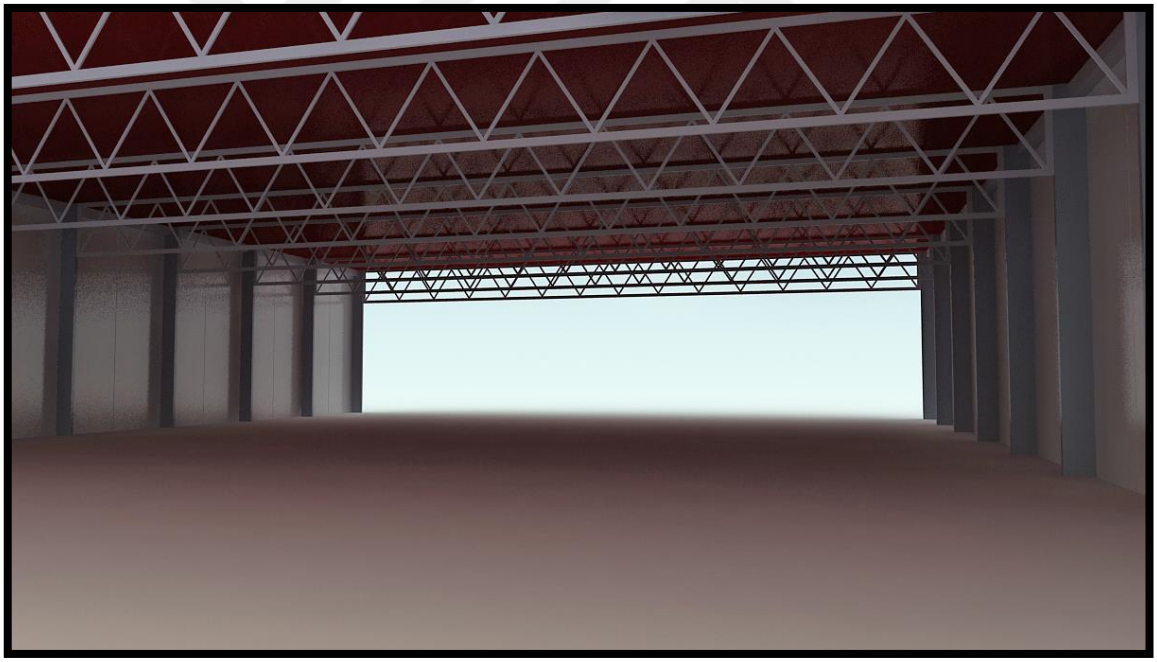
Şekil 3.18 Paralel başlıklı howe kafes kiriş görüntüsü – dış perspektif



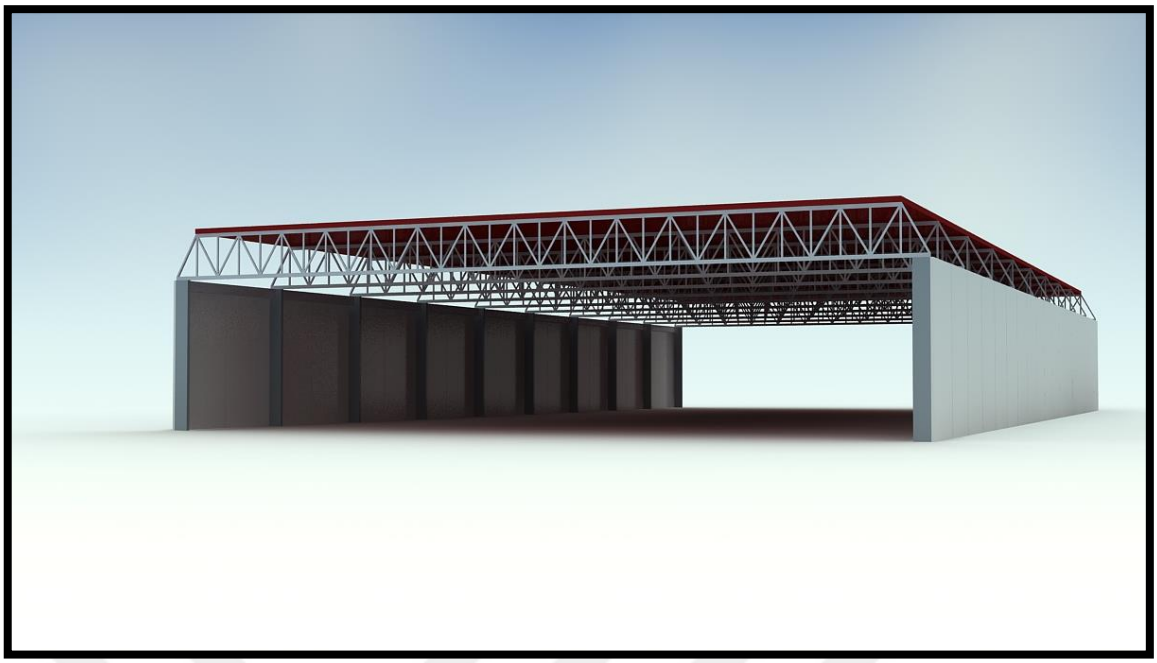
Şekil 3.19 Paralel başlıklı howe kafes kiriş görüntüsü – iç perspektif



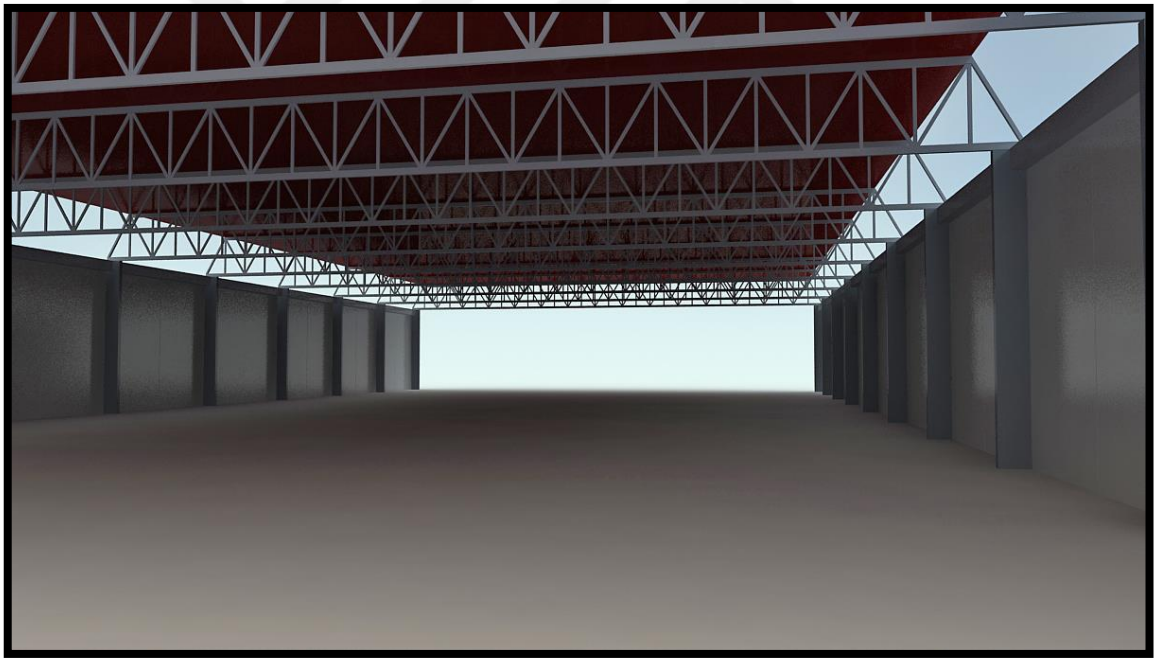
Şekil 3.20 Paralel başlıklı warren-simple kafes kiriş görüntüsü – dış perspektif



Şekil 3.21 Paralel başlıklı warren-simple kafes kiriş görüntüsü – iç perspektif



Şekil 3.22 Paralel başlıklı warren-modified kafes kiriş görüntüsü – dış perspektif



Şekil 3.23 Paralel başlıklı warren-modified kafes kiriş görüntüsü – iç perspektif

BÖLÜM 4

SONUÇ VE ÖNERİLER

Bu tez çalışmasında geniş açıklıklı taşıyıcı sistemlerin tasarımı ile ilgili bir yardımcı bilgisayar yazılımı oluşturulmuştur. PSS adı verilen yazılım hem tasarımda hem de eğitimde kullanılma potansiyeline sahip olup, hem işlevselliğinin hem de kapsamının ileri çalışmalarla daha da geliştirilmesi mümkündür.

PSS, mimarlık, yapı fiziği ve parametrik modelleme alanlarından ilgili konuların incelenmesi, birleştirilmesi ve harmanlanmasıyla oluşturulmuştur. Tez çalışması bir yandan PSS yazılımının özelliklerini ve nasıl kullanıldığını açıklarken diğer yandan MaxScript ve parametrik tasarım hakkında genel bilgiler aktararak bu ikisinin karşılıklı olarak ne kadar ilişkili olduğunu ortaya koymaktadır.

PSS, 3DS Max programı altında çalışan bir kod dizisidir. Bu kod dizisi kendi arayüzüne sahiptir. Dolayısı ile kullanıcı PSS'i kullanabilmek için 3DS Max programına hakim olmak zorunda değildir. Kullanıcılar, gerekli bilgilendirmelerle ve resimli düğmelerle oluşturulmuş arayüzü zorlanmadan kullanabilmektedir. Geniş açıklıklı taşıyıcı sistem modellerinin uzun ve zahmetli çizim ve hesaplama süreçleri PSS sayesinde birkaç adımda hızla ve doğru biçimde tamamlanabilmektedir. Oluşturulan modeller başka programlarda da kullanılmak üzere farklı formatlarda kaydedilebilmektedir.

Bu tez çalışması kapsamında üretilen yazılımda daha çok kafes sistemler üzerine ağırlık verilmiş ve mümkün olabildiğince çok kafes sistem tipi yazılım ortamına aktarılmaya çalışılmıştır. İleri bir çalışmada diğer taşıyıcı türlerinin de aynı veya başka yazılım ortamlarına aktarılabileceği öngörülmektedir. PSS'in eğitimde kullanım potansiyelleri de ileri çalışmalarla incelenmelidir. Yazılım, hem içeriği itibariyle taşıyıcı sistemler konusunda bir ders materyali olarak kullanılabileceği, hem de bilgisayar destekli

tasasrım alanındaki derslerde öğrencilere mimari kodlama konusunda bir örnek teşkil edebileceği değerlendirilmektedir.

KAYNAKLAR

[1] Apak K.,(2003), Büyük Açıklıklı Çelik Yapı Sistemleri ve Uygulanan Örneklerin İncelenmesi, Yüksek Lisans Tezi, MSÜ Fen Bilimleri Enstitüsü, İstanbul.

[2] Sungur İ.İ.., (2005), "Taşıyıcı Sistemler ve Yapı Statiği", Birsen Yayın Evi, İstanbul.

[3] Engel H., (2004), "Strüktür Sistemleri", Tasarım Yayın Grubu, İstanbul.

[4] Kolarevic, B., (2003), "Architecture In The Digital Age, Design and Manufacturing", Spoon Pres, NewYork.

[5] Akipek, Ö. F., (2004), Bilgisayar Teknolojilerinin Mimarlıkta Tasarım Gelistirme Amaçlı Kullanımları, Doktora Tezi, YTÜ Fen Bilimleri Enstitüsü, İstanbul.

[6] Hensel, M. ve Menges, A., (2008), Morpho-Ecologies, "Towards an Inclusive Discourse in Heterogenous Architectures, Architectural Association, AA Agendas, No.4, Londra.

[7] Terzi N.,(2009), Mimarlıkta Hesaplamalı Teknolojiler ve Geometri, Yüksek Lisans Tezi, YTÜ Fen Bilimleri Enstitüsü, İstanbul.

[8] Burry, M. ve Murray Z., (1997), "Computer-Aided Architectural Design Using Parametric Variation And Associative Geometry", 15th ECAADE-Conference Proceedings, Vienna.

[9] Tong T., (2000), Cephe Tipolojilerine Bağlı Olarak Geliştirilen Uzman Sistem Modeli, Doktora Tezi, YTÜ Fen Bilimleri Enstitüsü, İstanbul.

[10] GrafCad, Autocad, http://www.grafcad.com/autodesk_autocad2011.htm, 21 Kasım 2015

[11] Softpedia, One of the most powerful CAD applications, http://i1-win.softpedia-static.com/screenshots/AutoCAD_3.png, 5 Şubat 2016

[12] FGA, Allplan, http://fga.com.tr/yazilimlar/allplan, 21 Kasım 2015

[13] Araesoft, Allplan, https://i.ytimg.com/vi/IUV1Yj8Uxt4/maxresdefault.jpg 5 Şubat 2016

[14] Graphisoft, Archicad, http://www.graphisoft.com/products/archicad/ 21 Kasım 2015

[15] Bilderholaden, Archicad, http://www.bilderhochladen.eu/i/8a9d4e4d4f42.jpg, 5 Şubat 2016

[16] SayısalGrafik, Revit, http://www.sayisalgrafik.com.tr/index2.html?/urunler/revit/index.html, 22 Kasım 2015

[17] Typepad, Revit, http://a7.typepad.com/6a019b034dd8b4970c01bb0831203f970d-pi 5 Şubat 2016

[18] Wikipedia, Generative Components http://en.wikipedia.org/wiki/Generative Components#cite_note-2, 22 Kasım 2015

[19] Bentley, Generative Components https://www.bentley.com/~/asset/4/5956.ashx, 8 Şubat 2016

[20] FGA, Lumion, http://fga.com.tr/yazilimlar/lumion/, 22 Kasım 2015

[21] Lumion3D, Lumion Skills, http://lumion3d.be/wp-content/uploads/2010/10/editor.jpg , 8 Şubat 2016

[22] Warescrack, Blender to Unity http://warescrack.com/wp-content/uploads/2015/11/Blender-To-Unity3D-Material-Texture-3.jpg, 22 Kasım 2015

[23] Unity3D, Unity Platforms, http://unity3d.com/unity/multiplatform, 8 Şubat 2016

[24] Wikipedia, Unreal Engine, https://tr.wikipedia.org/wiki/Unreal_Engine, 22 Kasım 2015

[25] Ayanoğlu M. (2006), Mimarlık Eğitiminde Üç Boyutlu Bilgisayar Oyunu Motorlarının Kullanımı, Yüksek Lisans Tezi, İTÜ Fen Bilimleri Enstitüsü, İstanbul.

[26] Gscept, Unreal, http://focus.gscept.com/2015ip5/files/2015/01/unrealprintscreen.jpg, 8 Şubat 2016

[27] Wikipedia, Autodesk Maya, http://tr.wikipedia.org/wiki/Autodesk_Maya, 28 Kasım 2015

[28] Prota Bilgisayar, Autodesk Maya, http://www.prota.com.tr/prota_bilgisayar/urunler/maya.html, 28 Kasım 2015

[29] Wikipedia, Rhinoceros 3D, http://tr.wikipedia.org/wiki/Rhinoceros_3D 28 Kasım 2015

[30] Visualarq, Rhino, http://www.visualarq.com/wp-content/uploads/sites/2/2012/02/Robie-3D-and-Plans-R5.jpg, 10 Şubat 2016

[31] Worpress, RhinoScript, https://workshopsfactory.files.wordpress.com/2008/08/diagram3-blog.jpg, 10 Şubat 2016

[32] Worpress, RhinoScript, https://buesking2417.files.wordpress.com/2013/04/grasshopper-screenshot.jpg, 10 Şubat 2016

[33] Cg Bilişim, Grashopper, http://www.cgibilisim.com/egitim/Rhinoceros-Grasshopper-ve-V-ray-ile-3D-Mimari-Egitim-Paketi_7.html, 8 Mart 2016

[34] Autodesk, 3ds Max, http://usa.autodesk.com/3ds-max/ , 17 Kasım 2015

[35] Autodesk, 3ds Max, http://static-dc.autodesk.net/content/dam/autodesk/www/products/autodesk-3dsmax/images/features-2016/support-new-iray-mental-ray-enhancements-large-1152x696.jpg, 17 Kasım 2015

[36] Işık U.,(2009), Aydınlatma Programlarının Gelişmiş Işık Simülatörleri İle Olan İlişkisinin Değerlendirilmesi, Y Lisans, YTÜ Fen Bilimleri Enstitüsü, İstanbul.

[37] Autodesk, 3ds Max, http://static-dc.autodesk.net/content/dam/autodesk/www/products/autodesk-3dsmax/images/features-2016/support-new-iray-max-screen-large-1152x696.jpg, 17 Kasım 2015

[38] Autodesk, (2009), "3ds Max MAXScript Essentials", Second Edition, 3-4.

[39] cg-creatives, Maxscript, http://newspics.3d-worxx.com/behindthescenes/reviews/3dsmax2008/maxscript_editor.gif, 10 Şubat 2016

[40] Aksoy N., (2011), Max Script, http://nadiraksoy.com/MAXScriptKitabiGiris.html, 15 Ekim 2015

[41] Fao, Wide Span System Samples, http://www.fao.org/docrep/s1250e/S1250E4F.GIF, 15 Kasım 2015

[42] Chord Truss Systems, http://i01.i.aliimg.com/photo/v0/11895841/Parallel_Chord_Truss_Building_Systems.jpg, 15 Kasım 2015

[43] GÖRÜN, A., (2012), Çelik İskelet Yapıların Mimari Tasarımı. http://www.insaathaberleri.net/haber/275-celik-iskelet-yapilarin-mimaritasarimi, 22 Şubat 2016

[44] Wood Solutions, Wide Span Systems Details, http://www.timber.org.au/ewebeditpro4/upload/Arch_shapes_NAFI_b.jpg, 15 Kasım 2015

[45] Core, the olympic games, http://blog.core10.co.uk/img/greece_2008/129_olympic_site_arch.jpg , 23 Şubat 2016

[46] Sustainable Cities Vision, Evolution of Space Frames, http://sustainablecities vision. com/ 2015/07/09/evolution-of-space-frames/, 23 Mayıs 2016

[47] Gündoğ G.,(2007), Çağdaş Strüktür Sistemlerinin Mimarlığa Etkileri, Yüksek Lisans Tezi, Erciyes Ü. Fen Bilimleri Enstitüsü, Kayseri.

[48] Fiber Source, Textile Arch, http://www.fibersource.com/textile_Arch/Cat-2.htm, 15 Kasım 2015

[49] Pedersen S., (2008) Building Home, http://www.sonjapedersen.com/bcplace/bldghome_print-385.html, 15 Kasım 2015

[50] Türkçü H.Ç., (2003), "Çağdaş Taşıyıcı Sistemler", Birsen Yayın Evi, İstanbul.

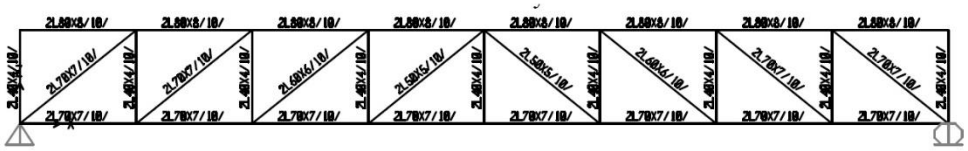
[51] DEREN, H., UZGİDER, E., PİROĞLU, F., ÇAĞLAYAN, Ö., (2008). "Çelik Yapılar", Çağlayan Kitabevi, İstanbul

[52] Çelik, K., (2012), Büyük Açıklıklı Yapılarda Çelik Taşıyıcılı Çatı Sistemleri Üzerine Bir İnceleme, Yüksek Lisans Tezi, Çukurova Ü. Fen Bilimleri Enstitüsü, Adana.

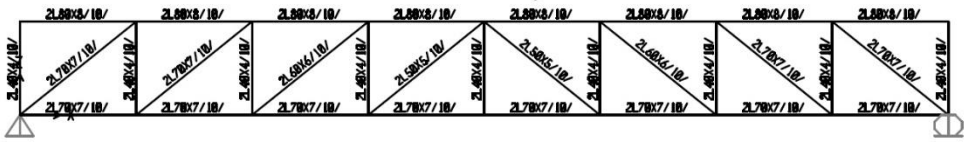
Kafes Sistem SAP2000 Çözümleri

Howe Tipi Kafes Kiriş Çözümleri

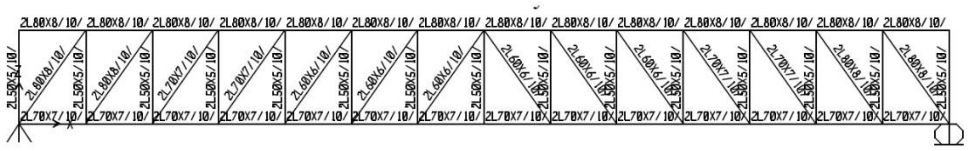
Paralel Başlık



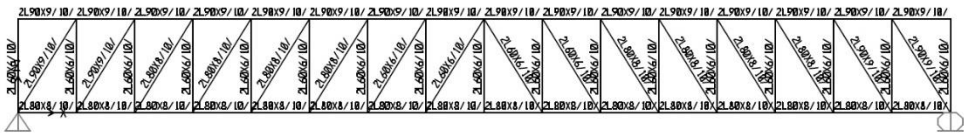
Kafes kiriş çubuklarının kesitleri
Howe tipi kafes sistemin SAP2000 çözümü (L = 15 m)



Kafes kiriş çubuklarının kesitleri
Howe tipi kafes sistemin SAP2000 çözümü (L = 20 m)



Kafes kiriş çubuklarının kesitleri
Howe tipi kafes sistemin SAP2000 çözümü (L = 25 m)

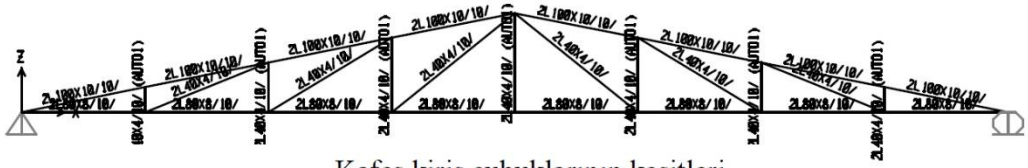


Kafes kiriş çubuklarının kesitleri
Howe tipi kafes sistemin SAP2000 çözümü (L = 30 m)

Howe Tipi Paralel Başlık Kafes Sistem SAP2000 Çözümü

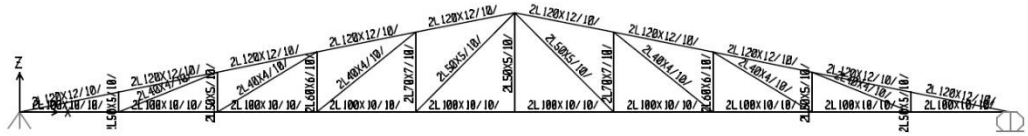
Howe Tipi Kafes Kiriş Çözümleri

Üçgen Başlık



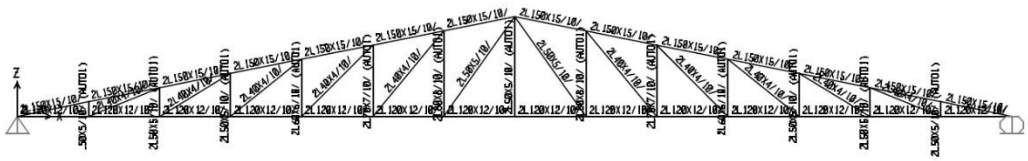
Kafes kiriş çubuklarının kesitleri

Howe tipi kafes sistemin SAP2000 çözümü (L = 15 m)



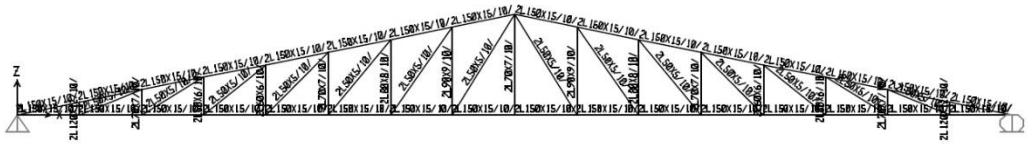
Kafes kiriş çubuklarının kesitleri

Howe tipi kafes sistemin SAP2000 çözümü (L = 20 m)



Kafes kiriş çubuklarının kesitleri

Howe tipi kafes sistemin SAP2000 çözümü (L = 25 m)



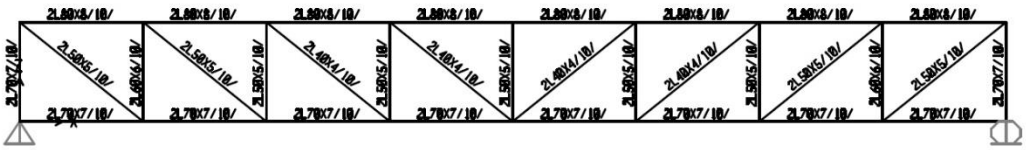
Kafes kiriş çubuklarının kesitleri

Howe tipi kafes sistemin SAP2000 çözümü (L = 30 m)

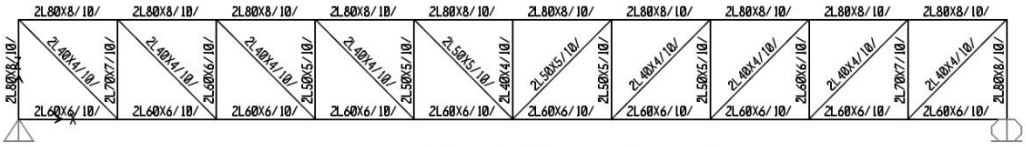
Howe Tipi Üçgen Başlık Kafes Sistem SAP2000 Çözümü

Pratt Tipi Kafes Kiriş Çözümleri

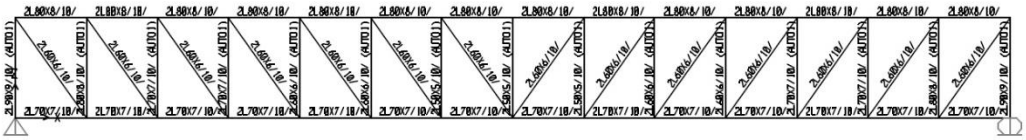
Paralel Başlık



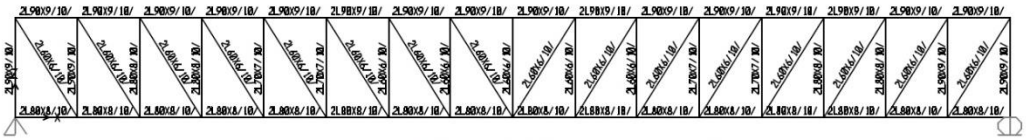
Kafes kiriş çubuklarının kesitleri
Pratt tipi kafes sistemin SAP2000 çözümü (L = 15 m)



Kafes kiriş çubuklarının kesitleri
Pratt tipi kafes sistemin SAP2000 çözümü (L = 20 m)



Kafes kiriş çubuklarının kesitleri
Pratt tipi kafes sistemin SAP2000 çözümü (L = 25 m)

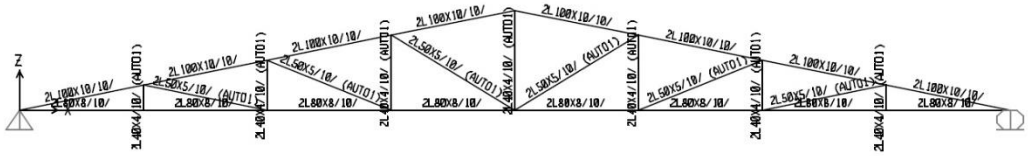


Kafes kiriş çubuklarının kesitleri
Pratt tipi kafes sistemin SAP2000 çözümü (L = 30 m)

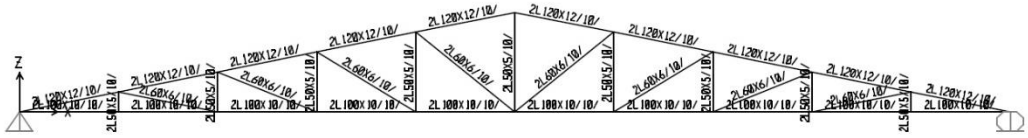
Pratt Tipi Paralel Başlık Kafes Sistem SAP2000 Çözümü

Pratt Tipi Kafes Kiriş Çözümleri

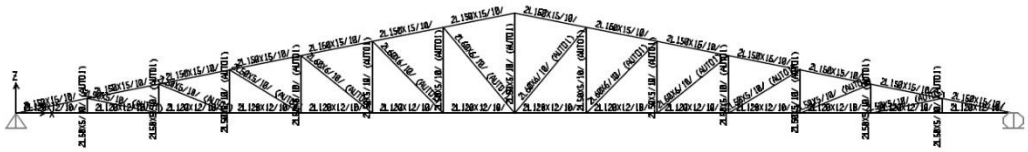
Üçgen Başlık



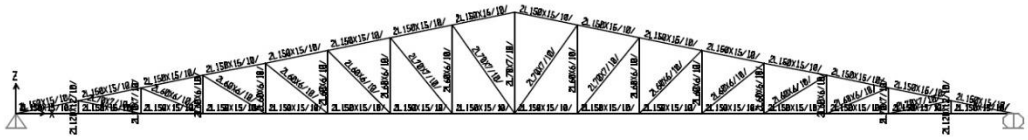
Kafes kiriş çubuklarının kesitleri
Pratt tipi kafes sistemin SAP2000 çözümü (L = 15 m)



Kafes kiriş çubuklarının kesitleri
Pratt tipi kafes sistemin SAP2000 çözümü (L = 20 m)



Kafes kiriş çubuklarının kesitleri
Pratt tipi kafes sistemin SAP2000 çözümü (L = 25 m)

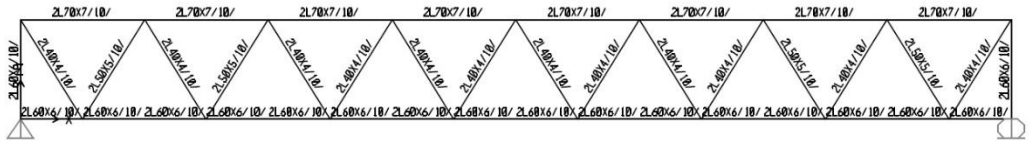


Kafes kiriş çubuklarının kesitleri
Pratt tipi kafes sistemin SAP2000 çözümü (L = 30 m)

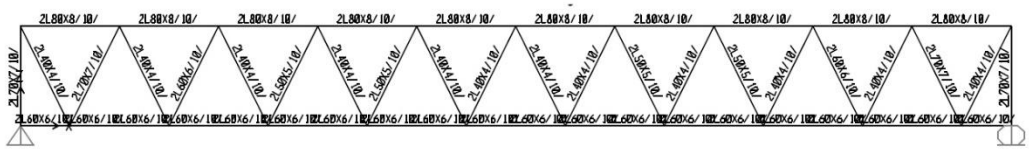
Pratt Tipi Üçgen Başlık Kafes Sistem SAP2000 Çözümü

Warren Tipi Kafes Kiriş Çözümleri

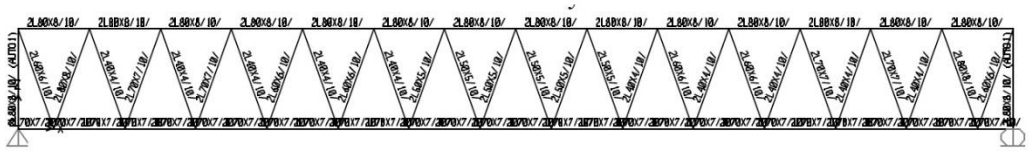
Paralel Başlık



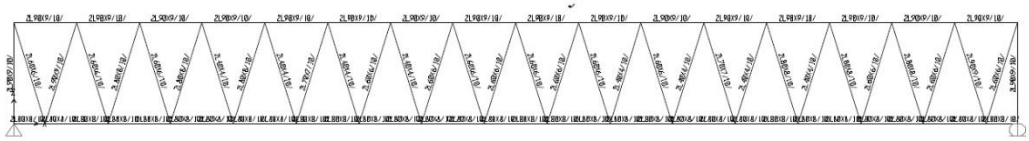
Kafes kiriş çubuklarının kesitleri
Warren tipi kafes sistemin SAP2000 çözümü (L = 15 m)



Kafes kiriş çubuklarının kesitleri
Warren tipi kafes sistemin SAP2000 çözümü (L = 20 m)



Kafes kiriş çubuklarının kesitleri
Warren tipi kafes sistemin SAP2000 çözümü (L = 25 m)

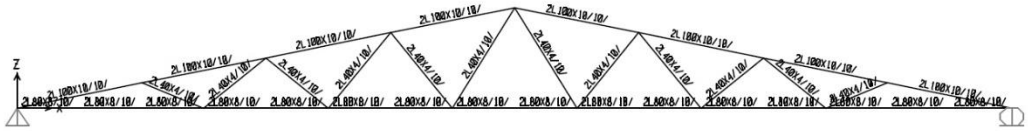


Kafes kiriş çubuklarının kesitleri
Warren tipi kafes sistemin SAP2000 çözümü (L = 30 m)

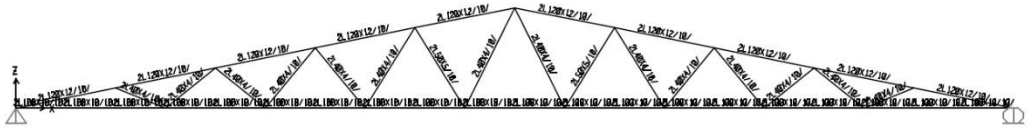
Warren Tipi Paralel Başlık Kafes Sistem SAP2000 Çözümü

Warren Tipi Kafes Kiriş Çözümleri

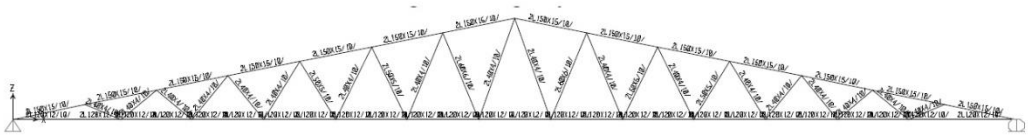
Üçgen Başlık



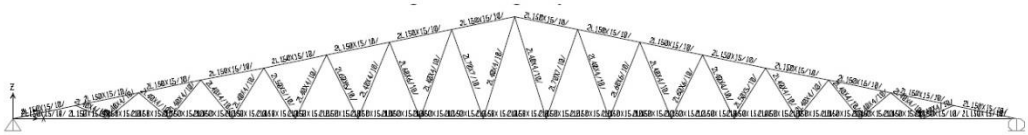
Kafes kiriş çubuklarının kesitleri
Warren tipi kafes sistemin SAP2000 çözümü (L = 15 m)



Kafes kiriş çubuklarının kesitleri
Warren tipi kafes sistemin SAP2000 çözümü (L = 20 m)



Kafes kiriş çubuklarının kesitleri
Warren tipi kafes sistemin SAP2000 çözümü (L = 25 m)



Kafes kiriş çubuklarının kesitleri
Warren tipi kafes sıstemin SAP2000 çözümü (L = 30 m)

Warren Tipi Üçgen Başlık Kafes Sistem SAP2000 Çözümü

EK-B

PSS Programı MaxScript Kodları

PSS.ms

```
PSS = NewRolloutFloater "P.S.S. 1.01 Beta" 240 770 800 100
filein "ParalelHOWE.ms"
filein "ParalelHOWE_Modified.ms"
filein "ParalelPRATT.ms"
filein "ParalelPRATT_Modified.ms"
filein "ParalelWARREN.ms"
filein "ParalelWARREN_Modified.ms"
filein "ParalelWARREN_Simple.ms"
filein "UcgenHowe.ms"
filein "PSS2016_Filein.ms"
filein "Exporting.ms"
--filein "Visualization.ms"
--include "Cameras.ms"
rollout SistemGorunus "Sistem Görünüsü" RolledUp:false
(
checkbox chbox_structure "Strüktür" checked:True enabled:false
checkbox chbox_duvar "Cephe Elemani Kullan" checked:True toolTip:"Olusacak Model için Duvar Seçenegi Sunar."
checkbox chbox_cati "Çati Örtüsü Kullan" checked:True toolTip:"Olusacak Model için Çati Seçenegi Sunar."
)
rollout RenderSistem "Render & Malzeme" RolledUp:false
(
group "Camera"
(
radiobuttons radbut_Cam_type labels:#("Camera Dis Mekan", "Camera ic Mekan") columns:1 align:#left toolTip:"Sisteminize Uygun Perspektifler Ayarlar ve Seçenek Sunar."
)
group "Render Sistem & Malzeme"
(
radiobuttons radbut_RenderType labels:#("Standart (default)", "Vray") columns:1 align:#left toolTip:"Sahnenin Malzeme ve Render Özelligini Belirlemenizi Saglar."
)
group "Render"
(
checkbox chbox_render "Render Al" checked:false toolTip:"Model Olustuktan Sonra Render Alir."
radiobuttons radbut_render_time labels:#("Sonra Hatirlat", "Sormadan Al") align:#left enabled:false default:2 toolTip:"Eger Render Al Seçili ise Size Seçenek Sunar."
)
on chbox_render changed state do radbut_render_time.enabled = state
)
--Global bilgiTipler, KafesKirisTipler, Sistemler, BilgiGenel, bilgiUcgen, UcgenKafesKiris, bilgiParalel, ParalelKafesKiris, bilgiPitch, PitchKafesKiris, bilgiBowstring, BowstringKafesKiris, bilgiParalelHowe, SisParaPlHowe, SistemGorunus, RenderSistem, bilgiUcgenHowe
bitmapAlphaIcon = bitmap 24 24 color:black
--Tüm sistemlerin görüldügü ve kisa bilgilendirmenin yer aldigi bölüm
rollout bilgiGenel "Genel Bilgi" RolledUp:true
(
group "Script Kullanimi"
(
label lab1 "--ilk panelde sistem tipi secilecektir." align:#left
label lab2 "--Secilen sisteme göre olusacak yeni panelden" align:#left
label lab3 "       bir kolon-kiris türü seçilmelidir." align:#left
label lab4 "--Son panelde ise sistem parametreleri ve" align:#left
label lab5 "       görüntüleme sekilleri belirlenmelidir." align:#left
)
)
rollout Sistemler "Sistem Çesitleri" RolledUp:false
(
group "Kiris Sistemler"
(
button buttDuz images:#("Truss_ScriptImages/KirisDuz.jpg", bitmapAlphaIcon, 1,1,1,1,1 ) width:200 height:100 align:#center
button buttDuz_txt "Düz Kirisler" border:true width:200 height:20 align:#center offset:[0,-4]
button buttKemer images:#("Truss_ScriptImages/KirisKemer.jpg", bitmapAlphaIcon, 1,1,1,1,1 ) width:200 height:100 align:#center offset:[0,10]
button buttKemer_txt "Kemer Kirisler" border:true width:200 height:20 align:#center offset:[0,-4]
button buttKafes images:#("Truss_ScriptImages/KirisKafes.jpg", bitmapAlphaIcon, 1,1,1,1,1 ) width:200 height:100 align:#center offset:[0,10]
button buttKafes_txt "Kafes(Makas) Kirisler" border:true width:200 height:20 align:#center offset:[0,-4]
button buttUzay images:#("Truss_ScriptImages/KirisUzay.jpg", bitmapAlphaIcon, 1,1,1,1,1 ) width:200 height:100 align:#center offset:[0,10]
button buttUzay_txt "Uzay Kafes Kirisler" border:true width:200 height:20 align:#center offset:[0,-4]
)
group "Özgün Tasarim"
(
button buttDesign images:#("Truss_ScriptImages/BenKendim.jpg", bitmapAlphaIcon, 1,1,1,1,1 ) width:200 height:100 align:#center
button buttDesign_txt "Ben Kendim Yaparim" border:true width:200 height:20 align:#center offset:[0,-4]
)
on buttKafes pressed do
(
addRollout bilgiTipler PSS
addRollout KafesKirisTipler PSS
removeRollout Sistemler PSS
removeRollout BilgiGenel PSS
)
on buttKafes_txt pressed do
(
addRollout bilgiTipler PSS
addRollout KafesKirisTipler PSS
removeRollout Sistemler PSS
removeRollout BilgiGenel PSS
)


)
 --Kafes Kiris Tiplerinin ve bilgilendirmesinin yer aldigi rolloutlar...
rollout bilgiTipler "Kisa Bilgi"  RolledUp:true
(
group "Script Kullanimi"
(
label lab1 "--Asagidaki baslik tiplerinden birini seçiniz." align:#left
)
)
rollout KafesKirisTipler "Kafes Kiris Baslik Tipleri" RolledUp:false
(
button buttGeri "<<< Üst Menüye Geri Dön <<<" border:true width:220 height:30 align:#center
group "Kafes Kiris (Truss) Tipleri"
(
button buttUcgen images:#("Truss_ScriptImages/TrussUcgen.jpg", bitmapAlphaIcon, 1,1,1,1,1 ) width:200 height:50 align:#center
button buttUcgen_txt "Üçgen Baslikli Kafes Kiris" border:true width:200 height:20 align:#center offset:[0,-4]
button buttParallel images:#("Truss_ScriptImages/TrussParallel.jpg", bitmapAlphaIcon, 1,1,1,1,1 ) width:200 height:50 align:#center offset:[0,2]
button buttParallel_txt "Paralel Baslikli Kafes Kiris" border:true width:200 height:20 align:#center offset:[0,-4]
button buttPitch images:#("Truss_ScriptImages/TrussPitch.jpg", bitmapAlphaIcon, 1,1,1,1,1 ) width:200 height:50 align:#center offset:[0,2]
button buttPitch_txt "Trapez(Pitch) Baslikli Kafes Kiris" border:true width:200 height:20 align:#center offset:[0,-4]
button buttBowstring images:#("Truss_ScriptImages/TrussBowstring.jpg", bitmapAlphaIcon, 1,1,1,1,1 ) width:200 height:50 align:#center offset:[0,2]
button buttBowstring_txt "Parabolik Baslikli Kafes Kiris" border:true width:200 height:20 align:#center offset:[0,-4]
)
on buttGeri pressed do
(
addRollout bilgiGenel PSS
addRollout Sistemler PSS
removeRollout bilgiTipler PSS
removeRollout KafesKirisTipler PSS
)
on buttUcgen pressed do
(
addRollout bilgiUcgen PSS
addRollout UcgenKafesKiris PSS
removeRollout bilgiTipler PSS
removeRollout KafesKirisTipler PSS
)
on buttUcgen_txt pressed do
(
addRollout bilgiUcgen PSS
addRollout UcgenKafesKiris PSS
removeRollout bilgiTipler PSS
removeRollout KafesKirisTipler PSS
)



on buttParallel pressed do
(
addRollout bilgiParalel PSS
addRollout ParalelKafesKiris PSS
removeRollout bilgiTipler PSS
removeRollout KafesKirisTipler PSS
)
on buttParallel_txt pressed do
(
addRollout bilgiParalel PSS
addRollout ParalelKafesKiris PSS
removeRollout bilgiTipler PSS
removeRollout KafesKirisTipler PSS
)
on buttPitch pressed do
(
addRollout bilgiPitch PSS
addRollout PitchKafesKiris PSS
removeRollout bilgiTipler PSS
removeRollout KafesKirisTipler PSS
)
on buttPitch_txt pressed do
(
addRollout bilgiPitch PSS
addRollout PitchKafesKiris PSS
```

```
removeRollout bilgiTipler PSS
removeRollout KafesKirisTipler PSS
)
on buttBowstring pressed do
(
addRollout bilgiBowstring PSS
addRollout BowstringKafesKiris PSS
removeRollout bilgiTipler PSS
removeRollout KafesKirisTipler PSS
)
on buttBowstring_txt pressed do
(
addRollout bilgiBowstring PSS
addRollout BowstringKafesKiris PSS
removeRollout bilgiTipler PSS
removeRollout KafesKirisTipler PSS
)
)
 --Üçgen baslikli Kafes Kiris Çesitlerinin ve bilgilendirmesinin yer aldigi rolloutlar
rollout bilgiUcgen "Kisa Bilgi"  RolledUp:true
(
group "Script Kullanimi"
(
label lab1 "--Asagidaki üçgen baslikli kafes kirislerden" align:#left
label lab2 "      birini seçiniz." align:#left
)
)
rollout UcgenKafesKiris "Üçgen Baslikli Kafes Kiris Baslik Çesitleri" RolledUp:false
(
button buttGeri "<<< Üst Menüye Geri Dön <<<" border:true width:220 height:30 align:#center
group "Üçgen Baslikli Kafes Kirisler"
(
button buttUcTrussHowe images:#("Truss_ScriptImages/TrussUcgen_Howe.jpg", bitmapAlphaIcon, 1,1,1,1,1 ) width:200 height:50 align:#center
button buttUcTrussHowe_txt "Kafes Kiris Howe (Gögüslemeli)" border:true width:200 height:20 align:#center offset:[0,-4]
button buttUcTrussPratt images:#("Truss_ScriptImages/TrussUcgen_Pratt.jpg", bitmapAlphaIcon, 1,1,1,1,1 ) width:200 height:50 align:#center offset:[0,2]
button buttUcTrussPratt_txt "Kafes Kiris Pratt (Ingiliz)" border:true width:200 height:20 align:#center offset:[0,-4]
button buttUcTrussFink images:#("Truss_ScriptImages/TrussUcgen_Fink.jpg", bitmapAlphaIcon, 1,1,1,1,1 ) width:200 height:50 align:#center offset:[0,2]
button buttUcTrussFink_txt "Kafes Kiris Fink" border:true width:200 height:20 align:#center offset:[0,-4]
button buttUcTrussFan images:#("Truss_ScriptImages/TrussUcgen_Fan.jpg", bitmapAlphaIcon, 1,1,1,1,1 ) width:200 height:50 align:#center offset:[0,2]
button buttUcTrussFan_txt "Kafes Kiris Fan" border:true width:200 height:20 align:#center offset:[0,-4]
button buttUcTrussScissors images:#("Truss_ScriptImages/TrussUcgen_Scissors.jpg", bitmapAlphaIcon, 1,1,1,1,1 ) width:200 height:50 align:#center offset:[0,2]
button buttUcTrussScissors_txt "Kafes Kiris Scissors" border:true width:200 height:20 align:#center offset:[0,-4]
button buttUcTrussPolonso images:#("Truss_ScriptImages/TrussUcgen_Polonso.jpg", bitmapAlphaIcon, 1,1,1,1,1 ) width:200 height:50 align:#center offset:[0,2]
button buttUcTrussPolonso_txt "Kafes Kiris Polonso" border:true width:200 height:20 align:#center offset:[0,-4]
button buttUcTrussPitchdual images:#("Truss_ScriptImages/TrussUcgen_PitchDual.jpg", bitmapAlphaIcon, 1,1,1,1,1 ) width:200 height:50 align:#center offset:[0,2]
button buttUcTrussPitchdual_txt "Kafes Kiris Pitch Dual" border:true width:200 height:20 align:#center offset:[0,-4]
button buttUcTrussShed images:#("Truss_ScriptImages/TrussUcgen_Shed.jpg", bitmapAlphaIcon, 1,1,1,1,1 ) width:200 height:50 align:#center offset:[0,2]
button buttUcTrussShed_txt "Kafes Kiris Shed" border:true width:200 height:20 align:#center offset:[0,-4]
button buttUcTrussPolynesian images:#("Truss_ScriptImages/TrussUcgen_Polynesian.jpg", bitmapAlphaIcon, 1,1,1,1,1 ) width:200 height:50 align:#center offset:[0,2]
button buttUcTrussPolynesian_txt "Kafes Kiris Polynesian" border:true width:200 height:20 align:#center offset:[0,-4]
button buttUcTrussInvertedDouble images:#("Truss_ScriptImages/TrussUcgen_InvertedDouble.jpg", bitmapAlphaIcon, 1,1,1,1,1 ) width:200 height:50 align:#center offset:[0,2]
button buttUcTrussInvertedDouble_txt "Kafes Kiris Inverted Double" border:true width:200 height:20 align:#center offset:[0,-4]
button buttUcTrussParallel images:#("Truss_ScriptImages/TrussUcgen_Parallel.jpg", bitmapAlphaIcon, 1,1,1,1,1 ) width:200 height:50 align:#center offset:[0,2]
button buttUcTrussParallel_txt "Kafes Kiris Paralel" border:true width:200 height:20 align:#center offset:[0,-4]
)
group "Mono-Üçgen Baslikli Kafes Kirisler"
(
button buttUcTrussMonoSc images:#("Truss_ScriptImages/TrussUcgen_MonoScissors.jpg", bitmapAlphaIcon, 1,1,1,1,1 ) width:200 height:50 align:#center
button buttUcTrussMonoSc_txt "Kafes Kiris Mono-Scissors" border:true width:200 height:20 align:#center offset:[0,-4]
button buttUcTrussMonoHowe images:#("Truss_ScriptImages/TrussUcgen_MonoHowe.jpg", bitmapAlphaIcon, 1,1,1,1,1 ) width:200 height:50 align:#center offset:[0,2]
button buttUcTrussMonoHowe_txt "Kafes Kiris Mono-Howe" border:true width:200 height:20 align:#center offset:[0,-4]
)
group "Hip-Üçgen Baslikli Kafes Kirisler"
(
button buttUcTrussHipGirder images:#("Truss_ScriptImages/TrussUcgen_HipGirder.jpg", bitmapAlphaIcon, 1,1,1,1,1 ) width:200 height:50 align:#center
button buttUcTrussHipGirder_txt "Kafes Kiris Hip Girder" border:true width:200 height:20 align:#center offset:[0,-4]
button buttUcTrussHipStep images:#("Truss_ScriptImages/TrussUcgen_HipStepdown.jpg", bitmapAlphaIcon, 1,1,1,1,1 ) width:200 height:50 align:#center offset:[0,2]
button buttUcTrussHipStep_txt "Kafes Kiris Hip Stepdown" border:true width:200 height:20 align:#center offset:[0,-4]
)
on buttGeri pressed do
(
addRollout bilgiTipler PSS
addRollout KafesKirisTipler PSS
removeRollout bilgiUcgen PSS
removeRollout UcgenKafesKiris PSS
)
on buttUcTrussHowe pressed do
(
addRollout bilgiUcgenHowe PSS
addRollout SisParaUcHowe PSS
addRollout SistemGorunus PSS
addRollout RenderSistem PSS
addRollout Exporting PSS
removeRollout bilgiUcgen PSS
removeRollout UcgenKafesKiris PSS
)
on buttUcTrussHowe_txt pressed do
(
addRollout bilgiUcgenHowe PSS
addRollout SisParaUcHowe PSS
addRollout SistemGorunus PSS
addRollout RenderSistem PSS
addRollout Exporting PSS
removeRollout bilgiUcgen PSS
removeRollout UcgenKafesKiris PSS
)
)
 --Paralel baslikli Kafes Kiris Çesitlerinin ve bilgilendirmesinin yer aldigi rolloutlar
rollout bilgiParalel "Kisa Bilgi"  RolledUp:true
(
group "Script Kullanimi"
(
label lab1 "--Asagidaki paralel baslikli kafes kirislerden" align:#left
label lab2 "      birini seçiniz." align:#left
)
)
rollout ParalelKafesKiris "Paralel Baslikli Kafes Kiris Baslik Çesitleri" RolledUp:false
(
button buttGeri "<<< Üst Menüye Geri Dön <<<" border:true width:220 height:30 align:#center
group "Paralel Baslikli Kafes Kirisler"
(
button buttPlTrussHowe images:#("Truss_ScriptImages/TrussParallel_Howe.jpg", bitmapAlphaIcon, 1,1,1,1,1 ) width:200 height:50 align:#center
button buttPlTrussHowe_txt "Kafes Kiris Howe" border:true width:200 height:20 align:#center offset:[0,-4]
button buttPlTrussPratt images:#("Truss_ScriptImages/TrussParallel_Pratt.jpg", bitmapAlphaIcon, 1,1,1,1,1 ) width:200 height:50 align:#center offset:[0,2]
button buttPlTrussPratt_txt "Kafes Kiris Pratt" border:true width:200 height:20 align:#center offset:[0,-4]
button buttPlTrussWarren images:#("Truss_ScriptImages/TrussParallel_Warren.jpg", bitmapAlphaIcon, 1,1,1,1,1 ) width:200 height:50 align:#center offset:[0,2]
button buttPlTrussWarren_txt "Kafes Kiris Warren" border:true width:200 height:20 align:#center offset:[0,-4]
button buttPlTrussWarrenS images:#("Truss_ScriptImages/TrussParallel_WarrenSimple.jpg", bitmapAlphaIcon, 1,1,1,1,1 ) width:200 height:50 align:#center offset:[0,2]
button buttPlTrussWarrenS_txt "Kafes Kiris Warren-Simple" border:true width:200 height:20 align:#center offset:[0,-4]
button buttPlTrussK images:#("Truss_ScriptImages/TrussParallel_K.jpg", bitmapAlphaIcon, 1,1,1,1,1 ) width:200 height:50 align:#center offset:[0,2]
button buttPlTrussK_txt "Kafes Kiris K" border:true width:200 height:20 align:#center offset:[0,-4]
button buttPlTrussKInv images:#("Truss_ScriptImages/TrussParallel_Kinverted.jpg", bitmapAlphaIcon, 1,1,1,1,1 ) width:200 height:50 align:#center offset:[0,2]
button buttPlTrussKInv_txt "Kafes Kiris K-Inverted" border:true width:200 height:20 align:#center offset:[0,-4]
)
group "Modified-Paralel Baslikli Kafes Kirisler"
(
```

```
button buttPlTrussHoweM images:#("Truss_ScriptImages/TrussParallel_ModifiedHowe.jpg", bitmapAlphaIcon, 1,1,1,1,1 ) width:200 height:50 align:#center
button buttPlTrussHoweM_txt "Kafes Kiris Howe-Modified" border:true width:200 height:20 align:#center offset:[0,-4]
button buttPlTrussPrattM images:#("Truss_ScriptImages/TrussParallel_ModifiedPratt.jpg", bitmapAlphaIcon, 1,1,1,1,1 ) width:200 height:50 align:#center offset:[0,2]
button buttPlTrussPrattM_txt "Kafes Kiris Pratt-Modified" border:true width:200 height:20 align:#center offset:[0,-4]
button buttPlTrussWarrenM images:#("Truss_ScriptImages/TrussParallel_ModifiedWarren.jpg", bitmapAlphaIcon, 1,1,1,1,1 ) width:200 height:50 align:#center offset:[0,2]
button buttPlTrussWarrenM_txt "Kafes Kiris Warren-Modified" border:true width:200 height:20 align:#center offset:[0,-4]
button buttPlTrussBalt images:#("Truss_ScriptImages/TrussParallel_Baltimore.jpg", bitmapAlphaIcon, 1,1,1,1,1 ) width:200 height:50 align:#center offset:[0,2]
button buttPlTrussBalt_txt "Kafes Kiris Baltimore" border:true width:200 height:20 align:#center offset:[0,-4]
)
on buttGeri pressed do
(
addRollout bilgiTipler PSS
addRollout KafesKirisTipler PSS
removeRollout bilgiParalel PSS
removeRollout ParalelKafesKiris PSS
)
on buttPlTrussHowe pressed do
(
addRollout bilgiParalelHowe PSS
addRollout SisParaPlHowe PSS
addRollout SistemGorunus PSS
addRollout RenderSistem PSS
addRollout Exporting PSS
removeRollout bilgiParalel PSS
removeRollout ParalelKafesKiris PSS
)
on buttPlTrussHowe_txt pressed do
(
addRollout bilgiParalelHowe PSS
addRollout SisParaPlHowe PSS
addRollout SistemGorunus PSS
addRollout RenderSistem PSS
addRollout Exporting PSS
removeRollout bilgiParalel PSS
removeRollout ParalelKafesKiris PSS
)
on buttPlTrussPratt pressed do
(
addRollout bilgiParalelPratt PSS
addRollout SisParaPlPratt PSS
addRollout SistemGorunus PSS
addRollout RenderSistem PSS
addRollout Exporting PSS
removeRollout bilgiParalel PSS
removeRollout ParalelKafesKiris PSS
)
on buttPlTrussPratt_txt pressed do
(
addRollout bilgiParalelPratt PSS
addRollout SisParaPlPratt PSS
addRollout SistemGorunus PSS
addRollout RenderSistem PSS
addRollout Exporting PSS
removeRollout bilgiParalel PSS
removeRollout ParalelKafesKiris PSS
)
on buttPlTrussWarren pressed do
(
addRollout bilgiParalelWarren PSS
addRollout SisParaPlWarren PSS
addRollout SistemGorunus PSS
addRollout RenderSistem PSS
addRollout Exporting PSS
removeRollout bilgiParalel PSS
removeRollout ParalelKafesKiris PSS
)

on buttPlTrussWarren_txt pressed do
(
addRollout bilgiParalelWarren PSS
addRollout SisParaPlWarren PSS
addRollout SistemGorunus PSS
addRollout RenderSistem PSS
addRollout Exporting PSS
removeRollout bilgiParalel PSS
removeRollout ParalelKafesKiris PSS
)
on buttPlTrussWarrenS pressed do
(
addRollout bilgiParalelWarrenS PSS
addRollout SisParaPlWarrenS PSS
addRollout SistemGorunus PSS
addRollout RenderSistem PSS
addRollout Exporting PSS
removeRollout bilgiParalel PSS
removeRollout ParalelKafesKiris PSS
)
on buttPlTrussWarrenS_txt pressed do
(
addRollout bilgiParalelWarrenS PSS
addRollout SisParaPlWarrenS PSS
addRollout SistemGorunus PSS
addRollout RenderSistem PSS
addRollout Exporting PSS
removeRollout bilgiParalel PSS
removeRollout ParalelKafesKiris PSS
)
on buttPlTrussHoweM pressed do
(
addRollout bilgiParalelHoweM PSS
addRollout SisParaPlHoweM PSS
addRollout SistemGorunus PSS
addRollout RenderSistem PSS
addRollout Exporting PSS
removeRollout bilgiParalel PSS
removeRollout ParalelKafesKiris PSS
)
on buttPlTrussHoweM_txt pressed do
(
addRollout bilgiParalelHoweM PSS
addRollout SisParaPlHoweM PSS
addRollout SistemGorunus PSS
addRollout RenderSistem PSS
addRollout Exporting PSS
removeRollout bilgiParalel PSS
removeRollout ParalelKafesKiris PSS
)



on buttPlTrussPrattM pressed do
(
addRollout bilgiParalelPrattM PSS
addRollout SisParaPlPrattM PSS
addRollout SistemGorunus PSS
addRollout RenderSistem PSS
addRollout Exporting PSS
removeRollout bilgiParalel PSS
removeRollout ParalelKafesKiris PSS
)
on buttPlTrussPrattM_txt pressed do
(
addRollout bilgiParalelPrattM PSS
addRollout SisParaPlPrattM PSS
addRollout SistemGorunus PSS
addRollout RenderSistem PSS
addRollout Exporting PSS
removeRollout bilgiParalel PSS
removeRollout ParalelKafesKiris PSS
)
on buttPlTrussWarrenM pressed do
(
addRollout bilgiParalelWarrenM PSS
addRollout SisParaPlWarrenM PSS
addRollout SistemGorunus PSS
addRollout RenderSistem PSS
addRollout Exporting PSS
removeRollout bilgiParalel PSS
removeRollout ParalelKafesKiris PSS
)
on buttPlTrussWarrenM_txt pressed do
(
addRollout bilgiParalelWarrenM PSS
addRollout SisParaPlWarrenM PSS
addRollout SistemGorunus PSS
addRollout RenderSistem PSS
addRollout Exporting PSS
removeRollout bilgiParalel PSS
removeRollout ParalelKafesKiris PSS
)
)
--Trapez(Pitch) baslikli Kafes Kiris Çesitlerinin ve bilgilendirmesinin yer aldigi rolloutlar
rollout bilgiPitch "Kisa Bilgi"  RolledUp:true
(
group "Script Kullanimi"
(
label lab1 "--Asagidaki Trapez baslikli kafes kirislerden" align:#left
label lab2 "      birini seçiniz." align:#left
)
)
rollout PitchKafesKiris "Trapez(Pitch) Baslikli K. K. Baslik Çesitleri" RolledUp:false
(
button buttGeri "<<< Üst Menüye Geri Dön <<<" border:true width:220 height:30 align:#center
group "Trapez(Pitch) Baslikli Kafes Kirisler"
```

```
(
button buttPiTrussHowe images:#("Truss_ScriptImages/TrussPitch_Howe.jpg", bitmapAlphaIcon, 1,1,1,1,1 ) width:200 height:50 align:#center
button buttPiTrussHowe_txt "Kafes Kiris Howe" border:true width:200 height:20 align:#center offset:[0,-4]
button buttPiTrussPratt images:#("Truss_ScriptImages/TrussPitch_Pratt.jpg", bitmapAlphaIcon, 1,1,1,1,1 ) width:200 height:50 align:#center offset:[0,2]
button buttPiTrussPratt_txt "Kafes Kiris Pratt" border:true width:200 height:20 align:#center offset:[0,-4]
button buttPiTrussWarren images:#("Truss_ScriptImages/TrussPitch_Warren.jpg", bitmapAlphaIcon, 1,1,1,1,1 ) width:200 height:50 align:#center offset:[0,2]
button buttPiTrussWarren_txt "Kafes Kiris Warren" border:true width:200 height:20 align:#center offset:[0,-4]
button buttPiTrussWarrenS images:#("Truss_ScriptImages/TrussPitch_WarrenSimple.jpg", bitmapAlphaIcon, 1,1,1,1,1 ) width:200 height:50 align:#center offset:[0,2]
button buttPiTrussWarrenS_txt "Kafes Kiris Warren-Simple" border:true width:200 height:20 align:#center offset:[0,-4]
)
group "Modified-Pitch(Mansard) Baslikli Kafes Kirisler"
(
button buttPiTrussHoweM images:#("Truss_ScriptImages/TrussPitch_ModifiedHowe.jpg", bitmapAlphaIcon, 1,1,1,1,1 ) width:200 height:50 align:#center
button buttPiTrussHoweM_txt "Kafes Kiris Howe-Modified(Mansard)" border:true width:200 height:20 align:#center offset:[0,-4]
button buttPiTrussPrattM images:#("Truss_ScriptImages/TrussPitch_ModifiedPratt.jpg", bitmapAlphaIcon, 1,1,1,1,1 ) width:200 height:50 align:#center offset:[0,2]
button buttPiTrussPrattM_txt "Kafes Kiris Pratt-Modified(Mansard)" border:true width:200 height:20 align:#center offset:[0,-4]
button buttPiTrussWarrenM images:#("Truss_ScriptImages/TrussPitch_ModifiedWarren.jpg", bitmapAlphaIcon, 1,1,1,1,1 ) width:200 height:50 align:#center offset:[0,2]
button buttPiTrussWarrenM_txt "Kafes Kiris Warren-Modified(Mansard)" border:true width:200 height:20 align:#center offset:[0,-4]
)
group "Mono-Pitch Baslikli Kafes Kirisler"
(
button buttPiTrussMonoWar images:#("Truss_ScriptImages/TrussPitch_MonoWarren.jpg", bitmapAlphaIcon, 1,1,1,1,1 ) width:200 height:50 align:#center
button buttPiTrussMonoWar_txt "Kafes Kiris Mono-Warren" border:true width:200 height:20 align:#center offset:[0,-4]
button buttPiTrussMonoWarS images:#("Truss_ScriptImages/TrussPitch_MonoWarrenSimple.jpg", bitmapAlphaIcon, 1,1,1,1,1 ) width:200 height:50 align:#center offset:[0,2]
button buttPiTrussMonoWarS_txt "Kafes Kiris Mono-Warren Simple" border:true width:200 height:20 align:#center offset:[0,-4]
)
on buttGeri pressed do
(
addRollout bilgiTipler PSS
addRollout KafesKirisTipler PSS
removeRollout bilgiPitch PSS
removeRollout PitchKafesKiris PSS
))
 --Parabolik(Bowstring) baslikli Kafes Kiris Çesitlerinin ve bilgilendirmesinin yer aldigi rolloutlar
rollout bilgiBowstring "Kisa Bilgi"  RolledUp:true
(
group "Script Kullanimi"
(
label lab1 "--Asagidaki Parabolik baslikli kafes kirislerden" align:#left
label lab2 "      birini seçiniz." align:#left
))
rollout BowstringKafesKiris "Parabolik(Bowstring) Baslikli K. K. Baslik Çesitleri" RolledUp:false
(
button buttGeri "<<< Üst Menüye Geri Dön <<<" border:true width:220 height:30 align:#center
group "Parabolik(Bowstring) Baslikli Kafes Kirisler"
(
button buttBoTrussHowe images:#("Truss_ScriptImages/TrussBowstring_Howe.jpg", bitmapAlphaIcon, 1,1,1,1,1 ) width:200 height:50 align:#center
button buttBoTrussHowe_txt "Kafes Kiris Howe" border:true width:200 height:20 align:#center offset:[0,-4]
button buttBoTrussPratt images:#("Truss_ScriptImages/TrussBowstring_Pratt.jpg", bitmapAlphaIcon, 1,1,1,1,1 ) width:200 height:50 align:#center offset:[0,2]
button buttBoTrussPratt_txt "Kafes Kiris Pratt" border:true width:200 height:20 align:#center offset:[0,-4]
button buttBoTrussWarren images:#("Truss_ScriptImages/TrussBowstring_Warren.jpg", bitmapAlphaIcon, 1,1,1,1,1 ) width:200 height:50 align:#center offset:[0,2]
button buttBoTrussWarren_txt "Kafes Kiris Warren" border:true width:200 height:20 align:#center offset:[0,-4]
button buttBoTrussWarrenS images:#("Truss_ScriptImages/TrussBowstring_WarrenSimple.jpg", bitmapAlphaIcon, 1,1,1,1,1 ) width:200 height:50 align:#center offset:[0,2]
button buttBoTrussWarrenS_txt "Kafes Kiris Warren-Simple" border:true width:200 height:20 align:#center offset:[0,-4]
)
group "Parabolik(Bowstring)-Double Baslikli Kafes Kirisler"
(
button buttBoTrussHoweD images:#("Truss_ScriptImages/TrussBowstring_DoubleHowe.jpg", bitmapAlphaIcon, 1,1,1,1,1 ) width:200 height:50 align:#center
button buttBoTrussHoweD_txt "Kafes Kiris Double-Howe" border:true width:200 height:20 align:#center offset:[0,-4]
button buttBoTrussPrattD images:#("Truss_ScriptImages/TrussBowstring_DoublePratt.jpg", bitmapAlphaIcon, 1,1,1,1,1 ) width:200 height:50 align:#center offset:[0,2]
button buttBoTrussPrattD_txt "Kafes Kiris Double-Pratt" border:true width:200 height:20 align:#center offset:[0,-4]
button buttBoTrussWarrenD images:#("Truss_ScriptImages/TrussBowstring_DoubleWarren.jpg", bitmapAlphaIcon, 1,1,1,1,1 ) width:200 height:50 align:#center offset:[0,2]
button buttBoTrussWarrenD_txt "Kafes Kiris Double-Warren" border:true width:200 height:20 align:#center offset:[0,-4]
button buttBoTrussWarrenDS images:#("Truss_ScriptImages/TrussBowstring_DoubleWarrenSimple.jpg", bitmapAlphaIcon, 1,1,1,1,1 ) width:200 height:50 align:#center offset:[0,2]
button buttBoTrussWarrenDS_txt "Kafes Kiris Double-Warren-Simple" border:true width:200 height:20 align:#center offset:[0,-4]
)
group "Lenticular Baslikli Kafes Kirisler"
(
button buttBoTrussHoweL images:#("Truss_ScriptImages/TrussBowstring_LenticularHowe.jpg", bitmapAlphaIcon, 1,1,1,1,1 ) width:200 height:50 align:#center
button buttBoTrussHoweL_txt "Kafes Kiris Lenticular Howe" border:true width:200 height:20 align:#center offset:[0,-4]
button buttBoTrussPrattL images:#("Truss_ScriptImages/TrussBowstring_LenticularPratt.jpg", bitmapAlphaIcon, 1,1,1,1,1 ) width:200 height:50 align:#center offset:[0,2]
button buttBoTrussPrattL_txt "Kafes Kiris Lenticular Pratt" border:true width:200 height:20 align:#center offset:[0,-4]
button buttBoTrussWarrenL images:#("Truss_ScriptImages/TrussBowstring_LenticularWarren.jpg", bitmapAlphaIcon, 1,1,1,1,1 ) width:200 height:50 align:#center offset:[0,2]
button buttBoTrussWarrenL_txt "Kafes Kiris Lenticular Warren" border:true width:200 height:20 align:#center offset:[0,-4]
)
group "Lenticular-Angled Baslikli Kafes Kirisler"
(
button buttBoTrussHoweLA images:#("Truss_ScriptImages/TrussBowstring_LenticularAngledHowe.jpg", bitmapAlphaIcon, 1,1,1,1,1 ) width:200 height:50 align:#center
button buttBoTrussHoweLA_txt "Kafes Kiris Lenticular Howe-Angled" border:true width:200 height:20 align:#center offset:[0,-4]
button buttBoTrussPrattLA images:#("Truss_ScriptImages/TrussBowstring_LenticularAngledPratt.jpg", bitmapAlphaIcon, 1,1,1,1,1 ) width:200 height:50 align:#center offset:[0,2]
button buttBoTrussPrattLA_txt "Kafes Kiris Lenticular Pratt-Angled" border:true width:200 height:20 align:#center offset:[0,-4]
button buttBoTrussWarrenLA images:#("Truss_ScriptImages/TrussBowstring_LenticularAngledWarren.jpg", bitmapAlphaIcon, 1,1,1,1,1 ) width:200 height:50 align:#center offset:[0,2]
button buttBoTrussWarrenLA_txt "Kafes Kiris Lenticular Warren-Angled" border:true width:200 height:20 align:#center offset:[0,-4]
)
on buttGeri pressed do
(
addRollout bilgiTipler PSS
addRollout KafesKirisTipler PSS
removeRollout bilgiBowstring PSS
removeRollout BowstringKafesKiris PSS
))
 AddRollout BilgiGenel PSS
AddRollout Sistemler PSS
```

KafesKirisTipleri.ms

```
rollout bilgiTipler "Kisa Bilgi"  RolledUp:true
(
group "Script Kullanimi"
(
label lab1 "--Asagidaki baslik tiplerinden birini seçiniz." align:#left
)
)
rollout KafesKirisTipler "Kafes Kiris Baslik Tipleri" RolledUp:false
(
button buttGeri "<<< Üst Menüye Geri Dön <<<" border:true width:220 height:30 align:#center
group "Kafes Kiris (Truss) Tipleri"
(
button buttUcgen images:#("Truss_ScriptImages/TrussUcgen.jpg", undefined, 1,1,1,1,1 ) width:200 height:50 align:#center
button buttUcgen_txt "Üçgen Baslikli Kafes Kiris" border:true width:200 height:20 align:#center offset:[0,-4]
button buttParallel images:#("Truss_ScriptImages/TrussParallel.jpg", undefined, 1,1,1,1,1 ) width:200 height:50 align:#center offset:[0,2]
button buttParallel_txt "Paralel Baslikli Kafes Kiris" border:true width:200 height:20 align:#center offset:[0,-4]
button buttPitch images:#("Truss_ScriptImages/TrussPitch.jpg", undefined, 1,1,1,1,1 ) width:200 height:50 align:#center offset:[0,2]
```

```
button buttPitch_txt "Trapez(Pitch) Baslikli Kafes Kiris" border:true width:200 height:20 align:#center offset:[0,-4]
button buttBowstring images:#("Truss_ScriptImages/TrussBowstring.jpg", undefined, 1,1,1,1,1 ) width:200 height:50 align:#center offset:[0,2]
button buttBowstring_txt "Parabolik Baslikli Kafes Kiris" border:true width:200 height:20 align:#center offset:[0,-4]
)
)
KKT = NewRolloutFloater "Kafes Kiris Baslik Tipleri" 240 770
AddRollout Bilgi KKT
AddRollout KafesKirisTipler KKT
```

KafesKirisUcgen.ms

```
rollout bilgiUcgen "Kisa Bilgi"  RolledUp:true
(
group "Script Kullanimi"
(
label lab1 "--Asagidaki üçgen baslikli kafes kirislerden" align:#left
label lab2 "      birini seçiniz." align:#left
)
)
rollout UcgenKafesKiris "Üçgen Baslikli Kafes Kiris Baslik Çesitleri" RolledUp:false
(
button buttGeri "<<< Üst Menüye Geri Dön <<<" border:true width:220 height:30 align:#center
group "Üçgen Baslikli Kafes Kirisler"
(
button buttUcTrussHowe images:#("Truss_ScriptImages/TrussUcgen_Howe.jpg", undefined, 1,1,1,1,1 ) width:200 height:50 align:#center
button buttUcTrussHowe_txt "Kafes Kiris Howe (Gögüslemeli)" border:true width:200 height:20 align:#center offset:[0,-4]
button buttUcTrussPratt images:#("Truss_ScriptImages/TrussUcgen_Pratt.jpg", undefined, 1,1,1,1,1 ) width:200 height:50 align:#center offset:[0,2]
button buttUcTrussPratt_txt "Kafes Kiris Pratt (Ingiliz)" border:true width:200 height:20 align:#center offset:[0,-4]
button buttUcTrussFink images:#("Truss_ScriptImages/TrussUcgen_Fink.jpg", undefined, 1,1,1,1,1 ) width:200 height:50 align:#center offset:[0,2]
button buttUcTrussFink_txt "Kafes Kiris Fink" border:true width:200 height:20 align:#center offset:[0,-4]
button buttUcTrussFan images:#("Truss_ScriptImages/TrussUcgen_Fan.jpg", undefined, 1,1,1,1,1 ) width:200 height:50 align:#center offset:[0,2]
button buttUcTrussFan_txt "Kafes Kiris Fan" border:true width:200 height:20 align:#center offset:[0,-4]
button buttUcTrussScissors images:#("Truss_ScriptImages/TrussUcgen_Scissors.jpg", undefined, 1,1,1,1,1 ) width:200 height:50 align:#center offset:[0,2]
button buttUcTrussScissors_txt "Kafes Kiris Scissors" border:true width:200 height:20 align:#center offset:[0,-4]
button buttUcTrussPolonso images:#("Truss_ScriptImages/TrussUcgen_Polonso.jpg", undefined, 1,1,1,1,1 ) width:200 height:50 align:#center offset:[0,2]
button buttUcTrussPolonso_txt "Kafes Kiris Polonso" border:true width:200 height:20 align:#center offset:[0,-4]

button buttUcTrussPitchdual images:#("Truss_ScriptImages/TrussUcgen_PitchDual.jpg", undefined, 1,1,1,1,1 ) width:200 height:50 align:#center offset:[0,2]
button buttUcTrussPitchdual_txt "Kafes Kiris Pitch Dual" border:true width:200 height:20 align:#center offset:[0,-4]
button buttUcTrussShed images:#("Truss_ScriptImages/TrussUcgen_Shed.jpg", undefined, 1,1,1,1,1 ) width:200 height:50 align:#center offset:[0,2]
button buttUcTrussShed_txt "Kafes Kiris Shed" border:true width:200 height:20 align:#center offset:[0,-4]
button buttUcTrussPolynesian images:#("Truss_ScriptImages/TrussUcgen_Polynesian.jpg", undefined, 1,1,1,1,1 ) width:200 height:50 align:#center offset:[0,2]
button buttUcTrussPolynesian_txt "Kafes Kiris Polynesian" border:true width:200 height:20 align:#center offset:[0,-4]
button buttUcTrussInvertedDouble images:#("Truss_ScriptImages/TrussUcgen_InvertedDouble.jpg", undefined, 1,1,1,1,1 ) width:200 height:50 align:#center offset:[0,2]
button buttUcTrussInvertedDouble_txt "Kafes Kiris Inverted Double" border:true width:200 height:20 align:#center offset:[0,-4]
button buttUcTrussParallel images:#("Truss_ScriptImages/TrussUcgen_Parallel.jpg", undefined, 1,1,1,1,1 ) width:200 height:50 align:#center offset:[0,2]
button buttUcTrussParallel_txt "Kafes Kiris Paralel" border:true width:200 height:20 align:#center offset:[0,-4]
)
group "Mono-Üçgen Baslikli Kafes Kirisler"
(
button buttUcTrussMonoSc images:#("Truss_ScriptImages/TrussUcgen_MonoScissors.jpg", undefined, 1,1,1,1,1 ) width:200 height:50 align:#center
button buttUcTrussMonoSc_txt "Kafes Kiris Mono-Scissors" border:true width:200 height:20 align:#center offset:[0,-4]
button buttUcTrussMonoHowe images:#("Truss_ScriptImages/TrussUcgen_MonoHowe.jpg", undefined, 1,1,1,1,1 ) width:200 height:50 align:#center offset:[0,2]
button buttUcTrussMonoHowe_txt "Kafes Kiris Mono-Howe" border:true width:200 height:20 align:#center offset:[0,-4]
)
group "Hip-Üçgen Baslikli Kafes Kirisler"
(
button buttUcTrussHipGirder images:#("Truss_ScriptImages/TrussUcgen_HipGirder.jpg", undefined, 1,1,1,1,1 ) width:200 height:50 align:#center
button buttUcTrussHipGirder_txt "Kafes Kiris Hip Girder" border:true width:200 height:20 align:#center offset:[0,-4]
button buttUcTrussHipStep images:#("Truss_ScriptImages/TrussUcgen_HipStepdown.jpg", undefined, 1,1,1,1,1 ) width:200 height:50 align:#center offset:[0,2]
button buttUcTrussHipStep_txt "Kafes Kiris Hip Stepdown" border:true width:200 height:20 align:#center offset:[0,-4]
)
)
UBKKC = NewRolloutFloater "Üçgen Baslikli Kafes Kiris Çesitleri" 240 770
AddRollout bilgiUcgen UBKKC
AddRollout UcgenKafesKiris UBKKC
```

KafesKirisParalel.ms

```
rollout bilgiParalel "Kisa Bilgi"  RolledUp:true
(
group "Script Kullanimi"
(
label lab1 "--Asagidaki paralel baslikli kafes kirislerden" align:#left
label lab2 "      birini seçiniz." align:#left
))
rollout ParalelKafesKiris "Paralel Baslikli Kafes Kiris Baslik Çesitleri" RolledUp:false
(
button buttGeri "<<< Üst Menüye Geri Dön <<<" border:true width:220 height:30 align:#center
group "Paralel Baslikli Kafes Kirisler"
(
button buttPlTrussHowe images:#("Truss_ScriptImages/TrussParallel_Howe.jpg", undefined, 1,1,1,1,1 ) width:200 height:50 align:#center
button buttPlTrussHowe_txt "Kafes Kiris Howe" border:true width:200 height:20 align:#center offset:[0,-4]
button buttPlTrussPratt images:#("Truss_ScriptImages/TrussParallel_Pratt.jpg", undefined, 1,1,1,1,1 ) width:200 height:50 align:#center offset:[0,2]
button buttPlTrussPratt_txt "Kafes Kiris Pratt" border:true width:200 height:20 align:#center offset:[0,-4]
button buttPlTrussWarren images:#("Truss_ScriptImages/TrussParallel_Warren.jpg", undefined, 1,1,1,1,1 ) width:200 height:50 align:#center offset:[0,2]
button buttPlTrussWarren_txt "Kafes Kiris Warren" border:true width:200 height:20 align:#center offset:[0,-4]
button buttPlTrussWarrenS images:#("Truss_ScriptImages/TrussParallel_WarrenSimple.jpg", undefined, 1,1,1,1,1 ) width:200 height:50 align:#center offset:[0,2]
button buttPlTrussWarrenS_txt "Kafes Kiris Warren-Simple" border:true width:200 height:20 align:#center offset:[0,-4]
button buttPlTrussK images:#("Truss_ScriptImages/TrussParallel_K.jpg", undefined, 1,1,1,1,1 ) width:200 height:50 align:#center offset:[0,2]
button buttPlTrussK_txt "Kafes Kiris K" border:true width:200 height:20 align:#center offset:[0,-4]
button buttPlTrussKInv images:#("Truss_ScriptImages/TrussParallel_Kinverted.jpg", undefined, 1,1,1,1,1 ) width:200 height:50 align:#center offset:[0,2]
button buttPlTrussKInv_txt "Kafes Kiris K-Inverted" border:true width:200 height:20 align:#center offset:[0,-4]
)
group "Modified-Paralel Baslikli Kafes Kirisler"
(
button buttPlTrussHoweM images:#("Truss_ScriptImages/TrussParallel_ModifiedHowe.jpg", undefined, 1,1,1,1,1 ) width:200 height:50 align:#center
button buttPlTrussHoweM_txt "Kafes Kiris Howe-Modified" border:true width:200 height:20 align:#center offset:[0,-4]
button buttPlTrussPrattM images:#("Truss_ScriptImages/TrussParallel_ModifiedPratt.jpg", undefined, 1,1,1,1,1 ) width:200 height:50 align:#center offset:[0,2]
button buttPlTrussPrattM_txt "Kafes Kiris Pratt-Modified" border:true width:200 height:20 align:#center offset:[0,-4]
button buttPlTrussWarrenM images:#("Truss_ScriptImages/TrussParallel_ModifiedWarren.jpg", undefined, 1,1,1,1,1 ) width:200 height:50 align:#center offset:[0,2]
button buttPlTrussWarrenM_txt "Kafes Kiris Warren-Modified" border:true width:200 height:20 align:#center offset:[0,-4]
button buttPlTrussBalt images:#("Truss_ScriptImages/TrussParallel_Baltimore.jpg", undefined, 1,1,1,1,1 ) width:200 height:50 align:#center offset:[0,2]
button buttPlTrussBalt_txt "Kafes Kiris Baltimore" border:true width:200 height:20 align:#center offset:[0,-4]
)
)
PBKKC = NewRolloutFloater "Paralel Baslikli Kafes Kiris Çesitleri" 240 770
AddRollout bilgiParalel PBKKC
AddRollout ParalelKafesKiris PBKKC
```

KafesKirisPitch.ms

```
rollout bilgiPitch "Kisa Bilgi"  RolledUp:true
(
group "Script Kullanimi"
(
label lab1 "--Asagidaki Trapez baslikli kafes kirislerden" align:#left
label lab2 "      birini seçiniz." align:#left
))
```

```
rollout PitchKafesKiris "Trapez(Pitch) Baslikli K. K. Baslik Çesitleri" RolledUp:false
(
button buttGeri "<<< Üst Menüye Geri Dön <<<" border:true width:220 height:30 align:#center
group "Trapez(Pitch) Baslikli Kafes Kirisler"
(
button buttPiTrussHowe images:#("Truss_ScriptImages/TrussPitch_Howe.jpg", undefined, 1,1,1,1,1 ) width:200 height:50 align:#center
button buttPiTrussHowe_txt "Kafes Kiris Howe" border:true width:200 height:20 align:#center offset:[0,-4]
button buttPiTrussPratt images:#("Truss_ScriptImages/TrussPitch_Pratt.jpg", undefined, 1,1,1,1,1 ) width:200 height:50 align:#center offset:[0,2]
button buttPiTrussPratt_txt "Kafes Kiris Pratt" border:true width:200 height:20 align:#center offset:[0,-4]
button buttPiTrussWarren images:#("Truss_ScriptImages/TrussPitch_Warren.jpg", undefined, 1,1,1,1,1 ) width:200 height:50 align:#center offset:[0,2]
button buttPiTrussWarren_txt "Kafes Kiris Warren" border:true width:200 height:20 align:#center offset:[0,-4]
button buttPiTrussWarrenS images:#("Truss_ScriptImages/TrussPitch_WarrenSimple.jpg", undefined, 1,1,1,1,1 ) width:200 height:50 align:#center offset:[0,2]
button buttPiTrussWarrenS_txt "Kafes Kiris Warren-Simple" border:true width:200 height:20 align:#center offset:[0,-4]
)
group "Modified-Pitch(Mansard) Baslikli Kafes Kirisler"
(
button buttPiTrussHoweM images:#("Truss_ScriptImages/TrussPitch_ModifiedHowe.jpg", undefined, 1,1,1,1,1 ) width:200 height:50 align:#center
button buttPiTrussHoweM_txt "Kafes Kiris Howe-Modified(Mansard)" border:true width:200 height:20 align:#center offset:[0,-4]
button buttPiTrussPrattM images:#("Truss_ScriptImages/TrussPitch_ModifiedPratt.jpg", undefined, 1,1,1,1,1 ) width:200 height:50 align:#center offset:[0,2]
button buttPiTrussPrattM_txt "Kafes Kiris Pratt-Modified(Mansard)" border:true width:200 height:20 align:#center offset:[0,-4]
button buttPiTrussWarrenM images:#("Truss_ScriptImages/TrussPitch_ModifiedWarren.jpg", undefined, 1,1,1,1,1 ) width:200 height:50 align:#center offset:[0,2]
button buttPiTrussWarrenM_txt "Kafes Kiris Warren-Modified(Mansard)" border:true width:200 height:20 align:#center offset:[0,-4]
)
group "Mono-Pitch Baslikli Kafes Kirisler"
(
button buttPiTrussMonoWar images:#("Truss_ScriptImages/TrussPitch_MonoWarren.jpg", undefined, 1,1,1,1,1 ) width:200 height:50 align:#center
button buttPiTrussMonoWar_txt "Kafes Kiris Mono-Warren" border:true width:200 height:20 align:#center offset:[0,-4]
button buttPiTrussMonoWarS images:#("Truss_ScriptImages/TrussPitch_MonoWarrenSimple.jpg", undefined, 1,1,1,1,1 ) width:200 height:50 align:#center offset:[0,2]
button buttPiTrussMonoWarS_txt "Kafes Kiris Mono-Warren Simple" border:true width:200 height:20 align:#center offset:[0,-4]
))
TBKKC = NewRolloutFloater "Trapez(Pitch) Baslikli Kafes Kiris Çesitleri" 240 770
AddRollout bilgiPitch TBKKC
AddRollout PitchKafesKiris TBKKC
```

KafesKirisParabolik.ms

```
rollout bilgiBowstring "Kisa Bilgi"  RolledUp:true
(
group "Script Kullanimi"
(
label lab1 "--Asagidaki Parabolik baslikli kafes kirislerden" align:#left
label lab2 "      birini seçiniz." align:#left
))
rollout BowstringKafesKiris "Parabolik(Bowstring) Baslikli K. K. Baslik Çesitleri" RolledUp:false
(
button buttGeri "<<< Üst Menüye Geri Dön <<<" border:true width:220 height:30 align:#center
group "Parabolik(Bowstring) Baslikli Kafes Kirisler"
(
button buttBoTrussHowe images:#("Truss_ScriptImages/TrussBowstring_Howe.jpg", undefined, 1,1,1,1,1 ) width:200 height:50 align:#center
button buttBoTrussHowe_txt "Kafes Kiris Howe" border:true width:200 height:20 align:#center offset:[0,-4]
button buttBoTrussPratt images:#("Truss_ScriptImages/TrussBowstring_Pratt.jpg", undefined, 1,1,1,1,1 ) width:200 height:50 align:#center offset:[0,2]
button buttBoTrussPratt_txt "Kafes Kiris Pratt" border:true width:200 height:20 align:#center offset:[0,-4]
button buttBoTrussWarren images:#("Truss_ScriptImages/TrussBowstring_Warren.jpg", undefined, 1,1,1,1,1 ) width:200 height:50 align:#center offset:[0,2]
button buttBoTrussWarren_txt "Kafes Kiris Warren" border:true width:200 height:20 align:#center offset:[0,-4]
button buttBoTrussWarrenS images:#("Truss_ScriptImages/TrussBowstring_WarrenSimple.jpg", undefined, 1,1,1,1,1 ) width:200 height:50 align:#center offset:[0,2]
button buttBoTrussWarrenS_txt "Kafes Kiris Warren-Simple" border:true width:200 height:20 align:#center offset:[0,-4]
)
group "Parabolik(Bowstring)-Double Baslikli Kafes Kirisler"
(
button buttBoTrussHoweD images:#("Truss_ScriptImages/TrussBowstring_DoubleHowe.jpg", undefined, 1,1,1,1,1 ) width:200 height:50 align:#center
button buttBoTrussHoweD_txt "Kafes Kiris Double-Howe" border:true width:200 height:20 align:#center offset:[0,-4]
button buttBoTrussPrattD images:#("Truss_ScriptImages/TrussBowstring_DoublePratt.jpg", undefined, 1,1,1,1,1 ) width:200 height:50 align:#center offset:[0,2]
button buttBoTrussPrattD_txt "Kafes Kiris Double-Pratt" border:true width:200 height:20 align:#center offset:[0,-4]
button buttBoTrussWarrenD images:#("Truss_ScriptImages/TrussBowstring_DoubleWarren.jpg", undefined, 1,1,1,1,1 ) width:200 height:50 align:#center offset:[0,2]
button buttBoTrussWarrenD_txt "Kafes Kiris Double-Warren" border:true width:200 height:20 align:#center offset:[0,-4]
button buttBoTrussWarrenDS images:#("Truss_ScriptImages/TrussBowstring_DoubleWarrenSimple.jpg", undefined, 1,1,1,1,1 ) width:200 height:50 align:#center offset:[0,2]
button buttBoTrussWarrenDS_txt "Kafes Kiris Double-Warren-Simple" border:true width:200 height:20 align:#center offset:[0,-4]
)
group "Lenticular Baslikli Kafes Kirisler"
(
button buttBoTrussHoweL images:#("Truss_ScriptImages/TrussBowstring_LenticularHowe.jpg", undefined, 1,1,1,1,1 ) width:200 height:50 align:#center
button buttBoTrussHoweL_txt "Kafes Kiris Lenticular Howe" border:true width:200 height:20 align:#center offset:[0,-4]

button buttBoTrussPrattL images:#("Truss_ScriptImages/TrussBowstring_LenticularPratt.jpg", undefined, 1,1,1,1,1 ) width:200 height:50 align:#center offset:[0,2]
button buttBoTrussPrattL_txt "Kafes Kiris Lenticular Pratt" border:true width:200 height:20 align:#center offset:[0,-4]
button buttBoTrussWarrenL images:#("Truss_ScriptImages/TrussBowstring_LenticularWarren.jpg", undefined, 1,1,1,1,1 ) width:200 height:50 align:#center offset:[0,2]
button buttBoTrussWarrenL_txt "Kafes Kiris Lenticular Warren" border:true width:200 height:20 align:#center offset:[0,-4]
)
group "Lenticular-Angled Baslikli Kafes Kirisler"
(
button buttBoTrussHoweLA images:#("Truss_ScriptImages/TrussBowstring_LenticularAngledHowe.jpg", undefined, 1,1,1,1,1 ) width:200 height:50 align:#center
button buttBoTrussHoweLA_txt "Kafes Kiris Lenticular Howe-Angled" border:true width:200 height:20 align:#center offset:[0,-4]
button buttBoTrussPrattLA images:#("Truss_ScriptImages/TrussBowstring_LenticularAngledPratt.jpg", undefined, 1,1,1,1,1 ) width:200 height:50 align:#center offset:[0,2]
button buttBoTrussPrattLA_txt "Kafes Kiris Lenticular Pratt-Angled" border:true width:200 height:20 align:#center offset:[0,-4]
button buttBoTrussWarrenLA images:#("Truss_ScriptImages/TrussBowstring_LenticularAngledWarren.jpg", undefined, 1,1,1,1,1 ) width:200 height:50 align:#center offset:[0,2]
button buttBoTrussWarrenLA_txt "Kafes Kiris Lenticular Warren-Angled" border:true width:200 height:20 align:#center offset:[0,-4]
))
BBKKC = NewRolloutFloater "Parabolik Baslikli Kafes Kiris Çesitleri" 240 770
AddRollout bilgiBowstring BBKKC
AddRollout BowstringKafesKiris BBKKC
```

ParalelHOWE.ms

```
--PSS = newRolloutFloater "P.S.S. 1.01 Beta" 240 770 800 100
GetFilePath = getThisScriptFilename()
SetFilePath= getFileNamePath GetFilePath
rollout bilgiParalelHowe "Kisa Bilgi"  RolledUp:true
(
group "Script Kullanimi"
(
label lab1 "--Asagidaki paralel baslikli HOWE kafes" align:#left
label lab2 "      kirisi sistem özelliklerinizi seçiniz ve" align:#left
label lab3 "      render - malzeme ayarlarini yapiniz." align:#left
))
rollout SisParaPlHowe "Sistem Parametreleri"
(
button buttGeri "<<< Üst Menüye Geri Dön <<<" border:true width:220 height:30 align:#center toolTip:"Bir Önceki Menüye Geri Döner"
group "Açikliklar"
(
spinner SistemGenis "Sistem Genisligi (cm):" range:[500,5000,2000] type:#float toolTip:"Sistemin Geçecegi Mesafeyi Belirler. (5m-50m)"
spinner SistemUzun "Sistem Uzunlugu (cm):" range:[500,30000,3000] type:#float  toolTip:"Yapinin Toplam Uzunlugunu Belirler. (5m-300m)"
)
group "Kolonlar"
(
spinner KolonYuks "Kolon Yüksekligi (cm):" range:[400,2000,800] type:#float toolTip:"Sistemin Yerden En Yüksek Noktasina Olan Mesafeyi Belirler. (4m-20m)"
spinner KolonSayisi "Kolon Sayisi:" range:[2,33,6] type:#integer toolTip:"Sistem Uzunlugu Yönündeki Toplam Aks Sayisini Belirler. (2-33)"
)
```

```
group "Diger Anlik Parametrik Bilgiler"
(
label labX1 "Kiris Yuksekligi: 125.0 cm" align:#left
label labX2 "Kirisin Yerden Yuksekligi: 675.0 cm" align:#left
label labX3 "Kolonlar Arasi Mesafe: 600.0 cm" align:#left
)
fn Paraminfo =
(
labX1.text = "Kiris Yuksekligi: " + (SistemGenis.value/16) as string + " cm"
labX2.text = "Kirisin Yerden Yuksekligi: " + (KolonYuks.value-(SistemGenis.value/16)) as string + " cm"
labX3.text = "Kolonlar Arasi Mesafe: " + ((SistemUzun.value)/(KolonSayisi.value-1)) as string + " cm"
)
on SistemGenis changed value do  Paraminfo()
on SistemUzun changed value do Paraminfo()
on KolonYuks changed value do Paraminfo()
on KolonSayisi changed value do Paraminfo()

button sil "Sil" border:true width:135 height:20 align:#left offset:[-10,0] across:2 toolTip:"Sahnedeki Bütün Nesneleri Siler."
on sil pressed do delete objects
button resett "Reset" border:true width:80 height:20 align:#right offset:[10,0] toolTip:"Sahneye Reset Atar (Yeniden Baslatir)."
on resett pressed do resetMaxFile()
button Olustur "Olustur" border:true width:220 height:30 align:#center offset:[0,0] toolTip:"Belirlediginiz Parametreler Dahilinde Sisteminizi Olusturur."
on Olustur pressed do
(
Global SisGen=SistemGenis.value--Sistem genisligi (Kolon merkezlerinden)
Global SisUz=SistemUzun.value--Sistem uzunlugu (Kolon merkezlerinden)
Global KolDist=SisUz/(KolonSayisi.value-1)--Kolonlar arasi mesafe (merkezlerinden)
Global KolSay=KolonSayisi.value--Sistemin tek siradaki kolon sayisi
Global KolEni=SisGen/50--Kolon eni
Global KolKal=KolDist/40--Kolon kalinligi
Global KolYuks=KolonYuks.value--Kolon yuksekligi
Global KirYuks=SisGen/16--Kiris yuksekligi
Global KirGen=SisGen-KolEni--Kiris genisligi
Global KirAltH=KolYuks-KirYuks--Kirisin yerden yuksekligi
Global KirHoPrYuk=SisGen/200--Kiris yatay profil yükseklikleri(kalinliklari)
Global KirVePrEni=KirHoPrYuk/2--Kiris düsey profil enleri(kalinliklari)
Global KirVePrYuk=KirYuks-(2*KirHoPrYuk)--Kiris düsey profil yükseklikleri
Global VerPrSayKS=(Pow (Pow SisGen 2) 0.2)/(Pow (Pow 1000 2) 0.2)--Kiris düsey profil sayisi KatSayisi (Sistem genisligi disinda, profillerin sikligini arttirip azaltmak için kullanilan bir parametre..)
Global VerPrSay=((((KirGen/2)-(KirHoPrYuk+(KirVePrEni/2)))/KirVePrYuk)*VerPrSayKS)-1--Kiris düsey profil sayisi nin yarisi
Global VerPrSay = (VerPrSay as integer)--Sayiyi tam sayi yaptik...
Global VerPrDist=((KirGen/2)-(KirHoPrYuk+(KirVePrEni/2))-(VerPrSay*KirVePrEni))/(VerPrSay+1)--Kiris düsey profiller arasi mesafe

Global DiagPrUz=sqrt ((Pow KirVePrYuk 2) + (Pow VerPrDist 2))--Kiris diagonal profil uzunlugu
Global DiagPrXPo=(DiagPrUz*(KirVePrEni/2))/KirVePrYuk--Kiris diagonal profil köseden X yönündeki mesafesi
Global DiagPrYpo=KirVePrYuk*DiagPrXPo/VerPrDist--Kiris diagonal profil köseden Y yönündeki mesafesi
Global theModi = extrude amount:(KolKal/4)
if (KolDist) > 1000 or (KolDist) < 300 then messagebox "DIKKAT!!! Girilen parametler uygun degil. Parametrelerinizi gözden geçiriniz... (Kolonlar arasi mesafe 300cm ile 1000cm arasinda olmali)"
else
(
if (KolYuks-KirYuks) < 300 then messagebox "DIKKAT!!! Girilen parametler uygun degildir. Parametrelerinizi gözden geçiriniz... (Kiris yerden yüksekligi minumum 300cm olmali)"
else
((
CamDis = Targetcamera fov:40 nearclip:1 farclip:1000 nearrange:0 farrange:1000 mpassEnabled:off mpassRenderPerPass:off name:"CameraDis"
$CameraDis.target = Targetobject transform:(matrix3 [1,0,0] [0,1,0] [0,0,1] [SisGen/2, SisUz/10, KolYuks/2])
$CameraDis.pos = [4*SisGen/3, -(2*((13*SisGen/5))/3), 200]
CamIc = Targetcamera fov:65 nearclip:1 farclip:1000 nearrange:0 farrange:1000 mpassEnabled:off mpassRenderPerPass:off name:"CameraIc"
$CameraIc.target = Targetobject transform:(matrix3 [1,0,0] [0,1,0] [0,0,1] [SisGen/2, 4*SisUz/5, KolYuks/2])
$CameraIc.pos = [5*SisGen/6, -500, 200]
)
case RenderSistem.radbut_RenderType.state of  --Render ve malzeme tipini seçer
(
1: (renderpresets.LoadAll 0 (SetFilePath + "Standard.rps")  -- (getFilenamePath (getSourceFileName()) + "SolidRocks.mse") Bu da baska bir örnek
--Steel1
meditMaterials[1] = Standardmaterial ()
meditMaterials[1].name = "Steel1"
meditMaterials[1].adLock = off
meditMaterials[1].ambient = color 80 80 80
meditMaterials[1].Diffuse = color 110 120 130
meditMaterials[1].specularLevel = 70
meditMaterials[1].glossiness = 50
--Steel2
meditMaterials[2] = Standardmaterial ()
meditMaterials[2].name = "Steel2"
meditMaterials[2].adLock = off
meditMaterials[2].ambient = color 80 80 80
meditMaterials[2].Diffuse = color 165 170 180
meditMaterials[2].specularLevel = 70
meditMaterials[2].glossiness = 50
--Roof
meditMaterials[7] = Standardmaterial ()
meditMaterials[7].name = "Roof"
meditMaterials[7].adLock = off
meditMaterials[7].ambient = color 80 80 80
meditMaterials[7].Diffuse = color 100 0 0
meditMaterials[7].specularLevel = 55
meditMaterials[7].glossiness = 30
--Panels
meditMaterials[8] = Standardmaterial ()
meditMaterials[8].name = "Panel"
meditMaterials[8].adLock = off
meditMaterials[8].ambient = color 80 80 80
meditMaterials[8].Diffuse = color 200 200 200
meditMaterials[8].specularLevel = 55
meditMaterials[8].glossiness = 30
--Ground
meditMaterials[13] = MatteShadow ()
meditMaterials[13].name = "Ground"
meditMaterials[13].opaqueAlpha = on
meditMaterials[13].applyAtmosphere = on
--BackGround için Vray
meditMaterials[14] = Standardmaterial ()
meditMaterials[14].name = "BackGroundVR"
meditMaterials[14].diffuseMap = Gradient ()
meditMaterials[14].diffuseMap.coordinates.mappingType = 1
meditMaterials[14].diffuseMap.coordinates.mapping = 3
meditMaterials[14].diffuseMap.name = "BG_Gradient"
meditMaterials[14].diffuseMap.color1 = color 0 72 255
meditMaterials[14].diffuseMap.color2 = color 201 223 223
meditMaterials[14].diffuseMap.color3 = color 255 254 237
meditMaterials[14].diffuseMap.map1 = Noise ()
meditMaterials[14].diffuseMap.map1.name = "BG-GR_Noise"
meditMaterials[14].diffuseMap.map1.color1 = color 0 62 154
meditMaterials[14].diffuseMap.map1.color2 = color 255 244 201
meditMaterials[14].diffuseMap.map1.size = 18
meditMaterials[14].diffuseMap.map1.phase = 7
--Environment Map ( BackGroundun diffuse unu kullaniyor)
environmentMap = Gradient ()
environmentMap = meditMaterials[14].diffuseMap
)
2: (renderpresets.LoadAll 0 (SetFilePath + "Vray.rps")  -- (getFilenamePath (getSourceFileName()) + "SolidRocks.mse") Bu da baska bir örnek
--Steel1
meditMaterials[1] = VRayMtl ()
meditMaterials[1].name = "Steel1"
meditMaterials[1].Diffuse = color 110 120 130
meditMaterials[1].Reflection = color 20 20 20
meditMaterials[1].reflection_lockGlossiness = off
meditMaterials[1].hilight_glossiness = 0.9
meditMaterials[1].reflection_glossiness = 0.9
--Steel2
meditMaterials[2] = VRayMtl ()
meditMaterials[2].name = "Steel2"
meditMaterials[2].Diffuse = color 165 170 180
meditMaterials[2].Reflection = color 20 20 20
meditMaterials[2].reflection_lockGlossiness = off
meditMaterials[2].hilight_glossiness = 0.9
meditMaterials[2].reflection_glossiness = 0.9
--Roof
meditMaterials[7] = VRayMtl ()
meditMaterials[7].name = "Roof"
meditMaterials[7].Diffuse = color 100 0 0
meditMaterials[7].Reflection = color 20 20 20
meditMaterials[7].reflection_lockGlossiness = off
meditMaterials[7].hilight_glossiness = 0.65
meditMaterials[7].reflection_glossiness = 0.9
--Panels
meditMaterials[8] = VRayMtl ()
meditMaterials[8].name = "Panel"
meditMaterials[8].Diffuse = color 200 200 200
meditMaterials[8].Reflection = color 20 20 20
meditMaterials[8].reflection_lockGlossiness = off
meditMaterials[8].hilight_glossiness = 0.65
meditMaterials[8].reflection_glossiness = 0.9
--Ground
meditMaterials[13] = VRayMtlWrapper ()
meditMaterials[13].name = "Ground"
meditMaterials[13].matteSurface = on
meditMaterials[13].matte_shadows = on
meditMaterials[13].matte_shadowsAffectAlpha = on
meditMaterials[13].baseMtl = VRayMtl ()
meditMaterials[13].baseMtl.name = "GroundVR"
meditMaterials[13].baseMtl.Diffuse = color 255 255 255
--BackGround için Vray
meditMaterials[14] = VRayMtl ()
```

```
meditMaterials[14].texmap_diffuse = Gradient ()
meditMaterials[14].texmap_diffuse.coordinates.mappingType = 1
meditMaterials[14].texmap_diffuse.coordinates.mapping = 3
meditMaterials[14].texmap_diffuse.color1 = color 0 72 255
meditMaterials[14].texmap_diffuse.color2 = color 201 223 223
meditMaterials[14].texmap_diffuse.color3 = color 255 254 237
meditMaterials[14].texmap_diffuse.noiseSize = 0
meditMaterials[14].texmap_diffuse.map1 = Noise ()
meditMaterials[14].texmap_diffuse.map1.color1 = color 0 62 154
meditMaterials[14].texmap_diffuse.map1.color2 = color 255 244 201
meditMaterials[14].texmap_diffuse.map1.size = 18
meditMaterials[14].texmap_diffuse.map1.phase = 7
meditMaterials[14].name = "BackGroundVR"
meditMaterials[14].texmap_diffuse.name = "BG_Gradient"
meditMaterials[14].texmap_diffuse.map1.name = "BG-GR_Noise"
--Environment Map ( BackGroundun diffuse unu kullaniyor)
environmentMap = Gradient ()
environmentMap = meditMaterials[14].texmap_diffuse
))
fn KirSist =
(
--Kolonlarin olusturulmasi
KolonSol = box width:KolEni length:KolKal height:KolYuks name: "KolonSol" wirecolor:gray
KolonSag = instance KolonSol name:"KolonSag" wirecolor:gray
Move KolonSag [SisGen, 0, 0]
--Kiris Olusturma
KirAlt = box width:KirGen length:(KolKal/2) height:KirHoPrYuk name: "KirisAltProfil" wirecolor:orange --Alt baslik
Move KirAlt [(SisGen/2), 0, KirAltH]
KirUst = instance KirAlt name: "KirisUstProfil" wirecolor:orange --Üst baslik
move KirUst [0, 0, (KirYuks-KirHoPrYuk)]
KirPrSol = box width:KirHoPrYuk length:(KolKal/2) height:KirVePrYuk name: "KirisSolProfil" wirecolor:orange --Baslik sol profil
Move KirPrSol [(KolEni/2)+(KirHoPrYuk/2), 0, KirAltH+KirHoPrYuk]
KirPrSag = instance KirPrSol name: "KirisSagProfil" wirecolor:orange --Baslik sag profil
Move KirPrSag [KirGen-KirHoPrYuk, 0, 0]
for i = 0 to (2*VerPrSay) do
(
KirPrVer = box width:KirVePrEni length:(KolKal/4) height:KirVePrYuk name: "KirisVerticalProfil" wirecolor:white --Vertical profiller
Move KirPrVer [((KolEni/2)+KirHoPrYuk+(KirVePrEni/2)+VerprDist), 0, KirAltH+KirHoPrYuk]
Move KirPrVer [(VerprDist+KirVePrEni)*i,0,0]
)
for m = 0 to VerPrSay do
(
DiagPrA = SplineShape pos:[0,0,0] name: "KirisDiagonalProfilA" --Diagonal Profiller
addNewSpline DiagPrA
addKnot DiagPrA 1 #corner #line [0,0,0]
addKnot DiagPrA 1 #corner #line [DiagPrXPo,0,0]
addKnot DiagPrA 1 #corner #line [VerPrDist,(KirVePrYuk-DiagPrYPo),0]
addKnot DiagPrA 1 #corner #line [VerPrDist,KirVePrYuk,0]
addKnot DiagPrA 1 #corner #line [(VerPrDist-DiagPrXPo),KirVePrYuk,0]
addKnot DiagPrA 1 #corner #line [0,DiagPrYPo,0]
Close DiagPrA 1
DiagPrA.wirecolor = color 255 255 255
rotate DiagPrA (eulerangles 90 0 0)
updateShape DiagPrA
Move DiagPrA [(KolEni/2)+KirHoPrYuk, (KolKal/8), KirAltH+KirHoPrYuk]
Move DiagPrA [(VerprDist+KirVePrEni)*m,0,0]
addmodifier DiagPrA theModi
)
for n = 0 to VerPrSay do
(
DiagPrB = SplineShape pos:[0,0,0] name: "KirisDiagonalProfilB" --Diagonal Profiller
addNewSpline DiagPrB
addKnot DiagPrB 1 #corner #line [0,0,0]
addKnot DiagPrB 1 #corner #line [-DiagPrXPo,0,0]
addKnot DiagPrB 1 #corner #line [-VerPrDist,(KirVePrYuk-DiagPrYPo),0]
addKnot DiagPrB 1 #corner #line [-VerPrDist,KirVePrYuk,0]
addKnot DiagPrB 1 #corner #line [-(VerPrDist-DiagPrXPo),KirVePrYuk,0]
addKnot DiagPrB 1 #corner #line [0,DiagPrYPo,0]
Close DiagPrB 1
DiagPrB.wirecolor = color 255 255 255
rotate DiagPrB (eulerangles 90 0 0)
updateShape DiagPrB
Move DiagPrB [(KolEni/2)+KirHoPrYuk+((VerprDist+KirVePrEni)*(VerPrSay+2))-KirVePrEni, (KolKal/8), KirAltH+KirHoPrYuk]
Move DiagPrB [(VerprDist+KirVePrEni)*n,0,0]
addmodifier DiagPrB theModi
)) --Fn KirSist Kapama
for t = 0 to KolSay-1 do--Kiris-Kolon sistemimizi burada tamamliyoruz ve tüm akslara kopyaliyoruz (instance olarak)
(
KirSist()
$kirisVerticalProfil*.baseobject = $kirisVerticalProfil*[1].baseobject
$KirisDiagonalProfilA*.baseobject = $KirisDiagonalProfilA*[1].baseobject
$KirisDiagonalProfilB*.baseobject = $KirisDiagonalProfilB*[1].baseobject
$KolonSol*.baseobject = $KolonSol*[1].baseobject
$KolonSag*.baseobject = $KolonSag*[1].baseobject
$KirisAltProfil*.baseobject = $KirisAltProfil*[1].baseobject
$KirisUstProfil*.baseobject = $KirisUstProfil*[1].baseobject
$KirisSolProfil*.baseobject = $KirisSolProfil*[1].baseobject
$KirisSagProfil*.baseobject = $KirisSagProfil*[1].baseobject
$KolonS*.material = meditMaterials[1]
$Kiris*.material = meditMaterials[2]
KirisSistem = group geometry name:"KirisSistem" select:false
Move KirisSistem [0, KolDist*t, 0]
)
for v = 0 to KolSay-2 do--Kiris sistemleri arasindaki düz profil krisleri yerlestirdik (yine instance olarak)
(
KirDuzSol = box width:(KolEni/2) length:(KolDist-KolKal) height:(KolDist/16) pos:[(-KolEni/4), (KolDist/2), (KolYuks-(KolDist/16))] name: "KirisDuzSol" Wirecolor:white
KirDuzSag = instance KirDuzSol name: "KirisDuzSag" Wirecolor:white
Move KirDuzSag [(Sisgen+(KolEni/2)), 0, 0]
Move KirDuzSol [0,KolDist*v,0]
Move KirDuzSag [0,KolDist*v,0]
$KirisDuzSol*.baseobject = $KirisDuzSol*[1].baseobject
$KirisDuzSag*.baseobject = $KirisDuzSag*[1].baseobject
$KirisDuz*.material = meditMaterials[2]
)
if SistemGorunus.chbox_duvar.checked then
(
for y = 0 to (2*KolSay)-5 do--Yan paneller - Ortadakiler (instance olarak)
(
PanelSol = box width:10 length:((KolDist/2)-2) height:KolYuks pos:[(-KolEni/2)-5, (KolDist/2)+(KolDist/4), 0] name: "PanelSol" Wirecolor:(color 100 150 200)
PanelSag = instance PanelSol name: "PanelSag" Wirecolor:(color 100 150 200)
Move PanelSag [(Sisgen+KolEni+10), 0, 0]
Move PanelSol [0,(KolDist/2)*y,0]
Move PanelSag [0,(KolDist/2)*y,0]
$PanelSol*.baseobject = $PanelSol*[1].baseobject
$PanelSag*.baseobject = $PanelSag*[1].baseobject
$Panel*.material = meditMaterials[8]
)

for z = 0 to 1 do--Yan paneller - Bastakiler (instance olarak)
(
PanelSol1 = box width:10 length:((KolDist/2)-2+(KolKal/2)+1) height:KolYuks pos:[(-KolEni/2)-5, (KolDist/4)-(KolKal/4)-0.5, 0] name: "PanelSol1" Wirecolor:(color 100 150 200)
PanelSag1 = instance PanelSol1 name: "PanelSag1" Wirecolor:(color 100 150 200)
Move PanelSag1 [(Sisgen+KolEni+10), 0, 0]
Move PanelSol1 [0,(SisUz-(KolDist/2)+8.5)*z,0]
Move PanelSag1 [0,(SisUz-(KolDist/2)+8.5)*z,0]
$PanelSol1*.baseobject = $PanelSol1*[1].baseobject
$PanelSag1*.baseobject = $PanelSag1*[1].baseobject
$Panel*.material = meditMaterials[8]
))
Zemin = plane width:(SisGen*3) length:(SisUz*3) pos:[(SisGen/2), (SisUz/2), 0] name:"Zemin" wirecolor:White --Zemin olusturur
$Zemin*.material = meditMaterials[13]
if SistemGorunus.chbox_cati.checked then --Çatiyi olusturur duruma göre
Cati = box width:(SisGen+KolEni+20) length:(SisUz+KolKal) height:20 pos:[(SisGen/2), (SisUz/2), KolYuks] name: "Cati" wirecolor:(color 150 0 0)
$Cati*.material = meditMaterials[7]
case RenderSistem.radbut_Cam_type.state of --Kameralari olusturur ve seçimini yapar
(
1: (select $CameraDis)
2: (select $CameraIc)
)
max vpt camera
deselect objects
backgroundColor = color 255 255 255
if RenderSistem.chbox_render.checked and (RenderSistem.radbut_render_time.state == 2) then --Render seçimini yapar ve render alir
max quick render
if (RenderSistem.radbut_render_time.state == 1) then --Render alir ama önce sorar
if queryBox "Sonra hatirlat demistin. Render alayim mi?" beep:false then
max quick render
) --If-Else kapama
) --If-Else kapama
) --On-do kapama
on buttGeri pressed do
(
addRollout bilgiParalel PSS
addRollout ParalelKafesKiris PSS
removeRollout bilgiParalelHowe PSS
removeRollout SisParaPlHowe PSS
removeRollout SistemGorunus PSS
removeRollout RenderSistem PSS
removeRollout Exporting PSS
)
) --Rollout kapama
--addRollout bilgiParalelHowe PSS
--addRollout SisParaPlHowe PSS
--addRollout SistemGorunus PSS
--addRollout RenderSistem PSS
```

ParalelHOWE_Modified.ms

```
--PSS = newRolloutFloater "P.S.S. 1.01 Beta" 240 770 800 100
GetFilePath = getThisScriptFilename()
SetFilePath= getFileNamePath GetFilePath
rollout bilgiParalelHoweM "Kisa Bilgi"  RolledUp:true
(
group "Script Kullanimi"
(
label lab1 "--Asagidaki paralel baslikli HOWE-Modified kafes" align:#left
label lab2 "       kirisi sistem özelliklerinizi seçiniz ve" align:#left
label lab3 "       render - malzeme ayarlarini yapiniz." align:#left
)) rollout SisParaPlHoweM "Sistem Parametreleri"
(
button buttGeri "<<< Üst Menüye Geri Dön <<<" border:true width:220 height:30 align:#center toolTip:"Bir Önceki Menüye Geri Döner"
group "Açikliklar"
(
spinner SistemGenis "Sistem Genisligi (cm):" range:[500,5000,2000] type:#float toolTip:"Sistemin Geçecegi Mesafeyi Belirler. (5m-50m)"
spinner SistemUzun "Sistem Uzunlugu (cm):" range:[500,30000,3000] type:#float  toolTip:"Yapinin Toplam Uzunlugunu Belirler. (5m-300m)"
)
group "Kolonlar"
(
spinner KolonYuks "Kolon Yüksekligi (cm):" range:[400,2000,800] type:#float toolTip:"Sistemin Yerden En Yüksek Noktasina Olan Mesafeyi Belirler. (4m-20m)"
spinner KolonSayisi "Kolon Sayisi:" range:[2,33,6] type:#integer toolTip:"Sistem Uzunlugu Yönündeki Toplam Aks Sayisini Belirler. (2-33)"
)
group "Diger Anlik Parametrik Bilgiler"
(
label labX1 "Kiris Yüksekligi: 125.0 cm" align:#left
label labX2 "Kirisin Yerden Yüksekligi: 675.0 cm" align:#left
label labX3 "Kolonlar Arasi Mesafe: 600.0 cm" align:#left
)
fn Paraminfo =
(
labX1.text = "Kiris Yüksekligi: " + (SistemGenis.value/16) as string + " cm"
labX2.text = "Kirisin Yerden Yüksekligi: " + (KolonYuks.value-(SistemGenis.value/16)) as string + " cm"
labX3.text = "Kolonlar Arasi Mesafe: " + ((SistemUzun.value)/(KolonSayisi.value-1)) as string + " cm"
)
on SistemGenis changed value do  Paraminfo()
on SistemUzun changed value do Paraminfo()
on KolonYuks changed value do Paraminfo()
on KolonSayisi changed value do Paraminfo()
button sil "Sil" border:true width:135 height:20 align:#left offset:[-10,0] across:2 toolTip:"Sahnedeki Bütün Nesneleri Siler."
on sil pressed do delete objects
button resett "Reset" border:true width:80 height:20 align:#right offset:[10,0] toolTip:"Sahneye Reset Atar (Yeniden Baslatir)."
on resett pressed do resetMaxFile()
button Olustur "Olustur" border:true width:220 height:30 align:#center offset:[0,0] toolTip:"Belirlediginiz Parametreler Dahilinde Sisteminizi Olusturur."
on Olustur pressed do
(
Global SisGen=SistemGenis.value--Sistem genisligi (Kolon merkezlerinden)
Global SisUz=SistemUzun.value--Sistem uzunlugu (Kolon merkezlerinden)
Global KolDist=SisUz/(KolonSayisi.value-1)--Kolonlar arasi mesafe (merkezlerinden)
Global KolSay=KolonSayisi.value--Sistemin tek siradaki kolon sayisi
Global KolEni=SisGen/50--Kolon eni
Global KolKal=KolDist/40--Kolon kalinligi
Global KolYuks=KolonYuks.value--Kolon yüksekligi
Global KirYuks=SisGen/16--Kiris yüksekligi
Global KirGen=SisGen+KolEni--Kiris genisligi
Global KirAltH=KolYuks-KirYuks--Kirisin yerden yüksekligi
Global KirHoPrYuk=SisGen/200--Kiris yatay profil yükseklikleri(kalinliklari)
Global KirVePrEni=KirHoPrYuk/2--Kiris düsey profil enleri(kalinliklari)
Global KirVePrYuk=KirYuks-(2*KirHoPrYuk)--Kiris düsey profil yükseklikleri
Global VerPrSayKS=(Pow (Pow SisGen 2) 0.2)/(Pow (Pow 1000 2) 0.2)--Kiris düsey profil sayisi KatSayisi (Sistem genisligi disinda, profillerin sikligini arttirip azaltmak için kullanilan bir parametre..)
Global VerPrSay=((((KirGen/2)-(KirVePrEni/2))/KirVePrYuk)*VerPrSayKS)-1--Kiris düsey profil sayisi nin yarisi
Global VerPrSay = (VerPrSay as integer)--Sayiyi tam sayi yaptik...
Global VerPrDist=((KirGen/2)-(KirVePrEni/2)-(VerPrSay*KirVePrEni))/(VerPrSay+1)--Kiris düsey profiller arasi mesafe
Global DiagPrUz=sqrt ((Pow KirVePrYuk 2) + (Pow VerPrDist 2))--Kiris diagonal profil uzunlugu
Global DiagPrXPo=(DiagPrUz*(KirVePrEni/2))/KirVePrYuk--Kiris diagonal profil köseden X yönündeki mesafesi
Global DiagPrYpo=KirVePrYuk*DiagPrXPo/VerPrDist--Kiris diagonal profil köseden Y yönündeki mesafesi
Global theModi = extrude amount:(KolKal/4)
Global theModi2 = extrude amount:(KolKal/2)
Global DiagPr1XPo=VerPrDist-((VerPrDist*KirVePrYuk)/(KirHoPrYuk+KirVePrYuk))--Kiris diagonal profil köseden X yönündeki mesafesi
if (KolDist) > 1000 or (KolDist) < 300 then messagebox "DIKKAT!!! Girilen parametler uygun degil. Parametrelerinizi gözden geçiriniz... (Kolonlar arasi mesafe 300cm ile 1000cm arasinda olmali)"
else
(
if (KolYuks-KirYuks) < 300 then messagebox "DIKKAT!!! Girilen parametler uygun degildir. Parametrelerinizi gözden geçiriniz... (Kiris yerden yüksekligi minumum 300cm olmali)"
else
((
CamDis = Targetcamera fov:40 nearclip:1 farclip:1000 nearrange:0 farrange:1000 mpassEnabled:off mpassRenderPerPass:off name:"CameraDis"
$CameraDis.target = Targetobject transform:(matrix3 [1,0,0] [0,1,0] [0,0,1] [SisGen/2, SisUz/10, KolYuks/2])
$CameraDis.pos = [4*SisGen/3, -(2*((13*SisGen/5))/3), 200]
CamIc = Targetcamera fov:65 nearclip:1 farclip:1000 nearrange:0 farrange:1000 mpassEnabled:off mpassRenderPerPass:off name:"CameraIc"
$CameraIc.target = Targetobject transform:(matrix3 [1,0,0] [0,1,0] [0,0,1] [SisGen/2, 4*SisUz/5, KolYuks/2])
$CameraIc.pos = [5*SisGen/6, -500, 200]
)
case RenderSistem.radbut_RenderType.state of  --Render ve malzeme tipini seçer
(
1: (renderpresets.LoadAll 0 (SetFilePath + "Standard.rps")

--Steel1
meditMaterials[1] = Standardmaterial ()
meditMaterials[1].name = "Steel1"
meditMaterials[1].adLock = off
meditMaterials[1].ambient = color 80 80 80
meditMaterials[1].Diffuse = color 110 120 130
meditMaterials[1].specularLevel = 70
meditMaterials[1].glossiness = 50
--Steel2
meditMaterials[2] = Standardmaterial ()
meditMaterials[2].name = "Steel2"
meditMaterials[2].adLock = off
meditMaterials[2].ambient = color 80 80 80
meditMaterials[2].Diffuse = color 165 170 180
meditMaterials[2].specularLevel = 70
meditMaterials[2].glossiness = 50
--Roof
meditMaterials[7] = Standardmaterial ()
meditMaterials[7].name = "Roof"
meditMaterials[7].adLock = off
meditMaterials[7].ambient = color 80 80 80
meditMaterials[7].Diffuse = color 100 0 0
meditMaterials[7].specularLevel = 55
meditMaterials[7].glossiness = 30
--Panels
meditMaterials[8] = Standardmaterial ()
meditMaterials[8].name = "Panel"
meditMaterials[8].adLock = off
meditMaterials[8].ambient = color 80 80 80
meditMaterials[8].Diffuse = color 200 200 200
meditMaterials[8].specularLevel = 55
meditMaterials[8].glossiness = 30
--Ground
meditMaterials[13] = MatteShadow ()
meditMaterials[13].name = "Ground"
meditMaterials[13].opaqueAlpha = on
meditMaterials[13].applyAtmosphere = on
--BackGround için Vray
meditMaterials[14] = Standardmaterial ()
meditMaterials[14].name = "BackGroundVR"
meditMaterials[14].diffuseMap = Gradient ()
meditMaterials[14].diffuseMap.coordinates.mappingType = 1
meditMaterials[14].diffuseMap.coordinates.mapping = 3
meditMaterials[14].diffuseMap.name = "BG_Gradient"
meditMaterials[14].diffuseMap.color1 = color 0 72 255
meditMaterials[14].diffuseMap.color2 = color 201 223 223
meditMaterials[14].diffuseMap.color3 = color 255 254 237
meditMaterials[14].diffuseMap.map1 = Noise ()
meditMaterials[14].diffuseMap.map1.name = "BG-GR_Noise"
meditMaterials[14].diffuseMap.map1.color1 = color 0 62 154
meditMaterials[14].diffuseMap.map1.color2 = color 255 244 201
meditMaterials[14].diffuseMap.map1.size = 18
meditMaterials[14].diffuseMap.map1.phase = 7
--Environment Map ( BackGroundun diffuse unu kullaniyor)
environmentMap = Gradient ()
environmentMap = meditMaterials[14].diffuseMap
)
2: (renderpresets.LoadAll 0 (SetFilePath + "Vray.rps")
--Steel1
meditMaterials[1] = VRayMtl ()
meditMaterials[1].name = "Steel1"
meditMaterials[1].Diffuse = color 110 120 130
meditMaterials[1].Reflection = color 20 20 20
meditMaterials[1].reflection_lockGlossiness = off
meditMaterials[1].hilight_glossiness = 0.9
meditMaterials[1].reflection_glossiness = 0.9
--Steel2
meditMaterials[2] = VRayMtl ()
meditMaterials[2].name = "Steel2"
meditMaterials[2].Diffuse = color 165 170 180
meditMaterials[2].Reflection = color 20 20 20
```

```
meditMaterials[2].reflection_lockGlossiness = off
meditMaterials[2].hilight_glossiness = 0.9
meditMaterials[2].reflection_glossiness = 0.9
--Roof
meditMaterials[7] = VRayMtl ()
meditMaterials[7].name = "Roof"
meditMaterials[7].Diffuse = color 100 0 0
meditMaterials[7].Reflection = color 20 20 20
meditMaterials[7].reflection_lockGlossiness = off
meditMaterials[7].hilight_glossiness = 0.65
meditMaterials[7].reflection_glossiness = 0.9
--Panels
meditMaterials[8] = VRayMtl ()
meditMaterials[8].name = "Panel"
meditMaterials[8].Diffuse = color 200 200 200
meditMaterials[8].Reflection = color 20 20 20
meditMaterials[8].reflection_lockGlossiness = off
meditMaterials[8].hilight_glossiness = 0.65
meditMaterials[8].reflection_glossiness = 0.9
--Ground
meditMaterials[13] = VRayMtlWrapper ()
meditMaterials[13].name = "Ground"
meditMaterials[13].matteSurface = on
meditMaterials[13].matte_shadows = on
meditMaterials[13].matte_shadowsAffectAlpha = on
meditMaterials[13].baseMtl = VRayMtl ()
meditMaterials[13].baseMtl.name = "GroundVR"
meditMaterials[13].baseMtl.Diffuse = color 255 255 255
--BackGround için Vray
meditMaterials[14] = VRayMtl ()
meditMaterials[14].texmap_diffuse = Gradient ()
meditMaterials[14].texmap_diffuse.coordinates.mappingType = 1
meditMaterials[14].texmap_diffuse.coordinates.mapping = 3
meditMaterials[14].texmap_diffuse.color1 = color 0 72 255
meditMaterials[14].texmap_diffuse.color2 = color 201 223 223
meditMaterials[14].texmap_diffuse.color3 = color 255 254 237
meditMaterials[14].texmap_diffuse.noiseSize = 0
meditMaterials[14].texmap_diffuse.map1 = Noise ()
meditMaterials[14].texmap_diffuse.map1.color1 = color 0 62 154
meditMaterials[14].texmap_diffuse.map1.color2 = color 255 244 201
meditMaterials[14].texmap_diffuse.map1.size = 18
meditMaterials[14].texmap_diffuse.map1.phase = 7
meditMaterials[14].name = "BackGroundVR"
meditMaterials[14].texmap_diffuse.name = "BG_Gradient"
meditMaterials[14].texmap_diffuse.map1.name = "BG-GR_Noise"
--Environment Map ( BackGroundun diffuse unu kullaniyor)
environmentMap = Gradient ()
environmentMap = meditMaterials[14].texmap_diffuse
))
fn KirSist =
(
--Kolonlarin olusturulmasi
KolonSol = box width:KolEni length:KolKal height:KirAltH name: "KolonSol" wirecolor:gray
KolonSag = instance KolonSol name:"KolonSag" wirecolor:gray
Move KolonSag [SisGen, 0, 0]
--Kiris Olusturma
KirAlt = box width:KirGen length:(KolKal/2) height:KirHoPrYuk name: "KirisAltProfil" wirecolor:orange --Alt baslik
Move KirAlt [(SisGen/2), 0, KirAltH]
KirUst = copy KirAlt name: "KirisUstProfil" wirecolor:orange --Üst baslik
KirUst.width = (KirGen-(2*VerPrDist))
move KirUst [0, 0, (KirYuks-KirHoPrYuk)]
for i = 0 to (2*VerPrSay) do
(
KirPrVer = box width:KirVePrEni length:(KolKal/4) height:KirVePrYuk name: "KirisVerticalProfil" wirecolor:white --Vertical profiller
Move KirPrVer [((KirVePrEni/2)+VerprDist-(KolEni/2)), 0, KirAltH+KirHoPrYuk]
Move KirPrVer [(VerprDist+KirVePrEni)*i,0,0]
)
DiagPrSol = SplineShape pos:[0,0,0] name: "KirisDiagonalProfilSolBas" --Diagonal Profil Sol Bastaki
addNewSpline DiagPrSol
addKnot DiagPrSol 1 #corner #line [0,0,0]
addKnot DiagPrSol 1 #corner #line [DiagPr1XPo,0,0]
addKnot DiagPrSol 1 #corner #line [VerPrDist,KirVePrYuk,0]
addKnot DiagPrSol 1 #corner #line [VerPrDist,KirVePrYuk+KirHoPrYuk,0]
Close DiagPrSol 1
DiagPrSol.wirecolor = orange
rotate DiagPrSol (eulerangles 90 0 0)
updateShape DiagPrSol
Move DiagPrSol [-(KolEni/2), (KolKal/4), KirAltH+KirHoPrYuk]
addmodifier DiagPrSol theModi2

DiagPrSag = SplineShape pos:[0,0,0] name: "KirisDiagonalProfilSagBas" --Diagonal Profil Sag Bastaki
addNewSpline DiagPrSag
addKnot DiagPrSag 1 #corner #line [0,0,0]
addKnot DiagPrSag 1 #corner #line [-DiagPr1XPo,0,0]
addKnot DiagPrSag 1 #corner #line [-VerPrDist,KirVePrYuk,0]
addKnot DiagPrSag 1 #corner #line [-VerPrDist,KirVePrYuk+KirHoPrYuk,0]
Close DiagPrSag 1
DiagPrSag.wirecolor = orange
rotate DiagPrSag (eulerangles 90 0 0)
updateShape DiagPrSag
Move DiagPrSag [KirGen-(KolEni/2), (KolKal/4), KirAltH+KirHoPrYuk]
addmodifier DiagPrSag theModi2
for m = 1 to VerPrSay do
(
DiagPrA = SplineShape pos:[0,0,0] name: "KirisDiagonalProfilA" --Diagonal Profiller
addNewSpline DiagPrA
addKnot DiagPrA 1 #corner #line [0,0,0]
addKnot DiagPrA 1 #corner #line [DiagPrXPo,0,0]
addKnot DiagPrA 1 #corner #line [VerPrDist,(KirVePrYuk-DiagPrYPo),0]
addKnot DiagPrA 1 #corner #line [VerPrDist,KirVePrYuk,0]
addKnot DiagPrA 1 #corner #line [(VerPrDist-DiagPrXPo),KirVePrYuk,0]
addKnot DiagPrA 1 #corner #line [0,DiagPrYPo,0]
Close DiagPrA 1
DiagPrA.wirecolor = color 255 255 255
rotate DiagPrA (eulerangles 90 0 0)
updateShape DiagPrA
Move DiagPrA [-(KolEni/2), (KolKal/8), KirAltH+KirHoPrYuk]
Move DiagPrA [(VerprDist+KirVePrEni)*m,0,0]
addmodifier DiagPrA theModi
)
for n = 0 to VerPrSay-1 do
(
DiagPrB = SplineShape pos:[0,0,0] name: "KirisDiagonalProfilB" --Diagonal Profiller
addNewSpline DiagPrB
addKnot DiagPrB 1 #corner #line [0,0,0]
addKnot DiagPrB 1 #corner #line [-DiagPrXPo,0,0]
addKnot DiagPrB 1 #corner #line [-VerPrDist,(KirVePrYuk-DiagPrYPo),0]
addKnot DiagPrB 1 #corner #line [-VerPrDist,KirVePrYuk,0]
addKnot DiagPrB 1 #corner #line [-(VerPrDist-DiagPrXPo),KirVePrYuk,0]
addKnot DiagPrB 1 #corner #line [0,DiagPrYPo,0]
Close DiagPrB 1
DiagPrB.wirecolor = color 255 255 255
rotate DiagPrB (eulerangles 90 0 0)
updateShape DiagPrB
Move DiagPrB [-(KolEni/2)+((VerprDist+KirVePrEni)*(VerPrSay+2))-KirVePrEni, (KolKal/8), KirAltH+KirHoPrYuk]
Move DiagPrB [(VerprDist+KirVePrEni)*n,0,0]
addmodifier DiagPrB theModi
)) --Fn KirSist Kapama

for t = 0 to KolSay-1 do--Kiris-Kolon sistemimizi burada tamamliyoruz ve tüm akslara kopyaliyoruz (instance olarak)
(
KirSist()
$kirisVerticalProfil*.baseobject = $kirisVerticalProfil*[1].baseobject
$KirisDiagonalProfilA*.baseobject = $KirisDiagonalProfilA*[1].baseobject
$KirisDiagonalProfilB*.baseobject = $KirisDiagonalProfilB*[1].baseobject
$KolonSol*.baseobject = $KolonSol*[1].baseobject
$KolonSag*.baseobject = $KolonSag*[1].baseobject
$KolonS*.material = meditMaterials[1]
$Kiris*.material = meditMaterials[2]
KirisSistem = group geometry name:"KirisSistem" select:false
Move KirisSistem [0, KolDist*t, 0]
)
for v = 0 to KolSay-2 do--Kiris sistemleri arasindaki düz profil krisleri yerlestirdik (yine instance olarak)
(
KirDuzSol = box width:(KolEni/2) length:(KolDist-KolKal) height:(KolDist/16) pos:[(-KolEni/4), (KolDist/2), (KirAltH-(KolDist/16))] name: "KirisDuzSol" Wirecolor:white
KirDuzSag = instance KirDuzSol name: "KirisDuzSag" Wirecolor:white
Move KirDuzSag [(Sisgen+(KolEni/2)), 0, 0]
Move KirDuzSol [0,KolDist*v,0]
Move KirDuzSag [0,KolDist*v,0]
$KirisDuzSol*.baseobject = $KirisDuzSol*[1].baseobject
$KirisDuzSag*.baseobject = $KirisDuzSag*[1].baseobject
$KirisDuz*.material = meditMaterials[2]
)
if SistemGorunus.chbox_duvar.checked then
(
for y = 0 to (2*KolSay)-5 do--Yan paneller - Ortadakiler (instance olarak)
(
PanelSol = box width:10 length:((KolDist/2)-2) height:KirAltH pos:[(-KolEni/2)-5, (KolDist/2)+(KolDist/4), 0] name: "PanelSol" Wirecolor:(color 100 150 200)
PanelSag = instance PanelSol name: "PanelSag" Wirecolor:(color 100 150 200)
Move PanelSag [(Sisgen+KolEni+10), 0, 0]
Move PanelSol [0,(KolDist/2)*y,0]
Move PanelSag [0,(KolDist/2)*y,0]
$PanelSol*.baseobject = $PanelSol*[1].baseobject
$PanelSag*.baseobject = $PanelSag*[1].baseobject
$Panel*.material = meditMaterials[8]
)
for z = 0 to 1 do--Yan paneller - Bastakiler (instance olarak)
(
PanelSol1 = box width:10 length:((KolDist/2)-2+(KolKal/2)+1) height:KirAltH pos:[(-KolEni/2)-5, (KolDist/4)-(KolKal/4)-0.5, 0] name: "PanelSol1" Wirecolor:(color 100 150 200)
PanelSag1 = instance PanelSol1 name: "PanelSag1" Wirecolor:(color 100 150 200)
Move PanelSag1 [(Sisgen+KolEni+10), 0, 0]
Move PanelSol1 [0,(SisUz-(KolDist/2)+8.5)*z,0]
```

```
Move PanelSag1 [0,(SisUz-(KolDist/2)+8.5)*z,0]
$PanelSol1*.baseobject = $PanelSol1*[1].baseobject
$PanelSag1*.baseobject = $PanelSag1*[1].baseobject
$Panel*.material = meditMaterials[8]
))
Zemin = plane width:(SisGen*3) length:(SisUz*3) pos:[(SisGen/2), (SisUz/2), 0] name:"Zemin" wirecolor:White --Zemin olusturur
$Zemin*.material = meditMaterials[13]
if SistemGorunus.chbox_cati.checked then --Çatiyi olusturur duruma göre
Cati = box width:(SisGen+KolEni-(2*VerPrDist)) length:(SisUz+KolKal) height:20 pos:[(SisGen/2), (SisUz/2), KolYuks] name: "Cati" wirecolor:(color 150 0 0)
$Cati*.material = meditMaterials[7]
case RenderSistem.radbut_Cam_type.state of --Kameralari olusturur ve seçimini yapar
(
1: (select $CameraDis)
2: (select $CameraIc)
)
max vpt camera
deselect objects
backgroundColor = color 255 255 255
if RenderSistem.chbox_render.checked and (RenderSistem.radbut_render_time.state == 2) then --Render seçimini yapar ve render alir
max quick render
if (RenderSistem.radbut_render_time.state == 1) then --Render alir ama önce sorar
if queryBox "Sonra hatirlat demistin. Render alayim mi?" beep:false then
max quick render
) --If-Else kapama
) --If-Else kapama
) --On-do kapama
on buttGeri pressed do
(
addRollout bilgiParalel PSS
addRollout ParalelKafesKiris PSS
removeRollout bilgiParalelHoweM PSS
removeRollout SisParaPlHoweM PSS
removeRollout SistemGorunus PSS
removeRollout RenderSistem PSS
removeRollout Exporting PSS
)
) --Rollout kapama--addRollout bilgiParalelHoweM PSS
--addRollout SisParaPlHoweM PSS
--addRollout SistemGorunus PSS
--addRollout RenderSistem PSS
```

ParalelPRATT.ms

```
--PSS = newRolloutFloater "P.S.S. 1.01 Beta" 240 770 800 100
GetFilePath = getThisScriptFilename()
SetFilePath= getFileNamePath GetFilePath
rollout bilgiParalelPratt "Kisa Bilgi" RolledUp:true
(
group "Script Kullanimi"
(
label lab1 "--Asagidaki paralel baslikli PRATT" align:#left
label lab2 " kafes kirisi sistem özelliklerinizi seçiniz" align:#left
label lab3 " ve render - malzeme ayarlarini yapiniz." align:#left
))
rollout SisParaPlPratt "Sistem Parametreleri"
(
button buttGeri "<<< Üst Menüye Geri Dön <<<" border:true width:220 height:30 align:#center toolTip:"Bir Önceki Menüye Geri Döner"
group "Açikliklar"
(
spinner SistemGenis "Sistem Genisligi (cm):" range:[500,5000,2000] type:#float toolTip:"Sistemin Geçecegi Mesafeyi Belirler. (5m-50m)"
spinner SistemUzun "Sistem Uzunlugu (cm):" range:[500,30000,3000] type:#float toolTip:"Yapinin Toplam Uzunlugunu Belirler. (5m-300m)"
)
group "Kolonlar"
(
spinner KolonYuks "Kolon Yüksekligi (cm):" range:[400,2000,800] type:#float toolTip:"Sistemin Yerden En Yüksek Noktasina Olan Mesafeyi Belirler. (4m-20m)"
spinner KolonSayisi "Kolon Sayisi:" range:[2,33,6] type:#integer toolTip:"Sistem Uzunlugu Yönündeki Toplam Aks Sayisini Belirler. (2-33)"
)
group "Diger Anlik Parametrik Bilgiler"
(
label labX1 "Kiris Yüksekligi: 125.0 cm" align:#left
label labX2 "Kirisin Yerden Yüksekligi: 675.0 cm" align:#left
label labX3 "Kolonlar Arasi Mesafe: 600.0 cm" align:#left
)
fn Paraminfo =
(
labX1.text = "Kiris Yüksekligi: " + (SistemGenis.value/16) as string + " cm"
labX2.text = "Kirisin Yerden Yüksekligi: " + (KolonYuks.value-(SistemGenis.value/16)) as string + " cm"
labX3.text = "Kolonlar Arasi Mesafe: " + ((SistemUzun.value)/(KolonSayisi.value-1)) as string + " cm"
)
on SistemGenis changed value do Paraminfo()
on SistemUzun changed value do Paraminfo()
on KolonYuks changed value do Paraminfo()
on KolonSayisi changed value do Paraminfo()
button sil "Sil" border:true width:135 height:20 align:#left offset:[-10,0] across:2 toolTip:"Sahnedeki Bütün Nesneleri Siler."
on sil pressed do delete objects
button resett "Reset" border:true width:80 height:20 align:#right offset:[10,0] toolTip:"Sahneye Reset Atar (Yeniden Baslatir)."
on resett pressed do resetMaxFile()
button Olustur "Olustur" border:true width:220 height:30 align:#center offset:[0,0] toolTip:"Belirlediginiz Parametreler Dahilinde Sisteminizi Olusturur."
on Olustur pressed do
(
Global SisGen=SistemGenis.value--Sistem genisligi (Kolon merkezlerinden)
Global SisUz=SistemUzun.value--Sistem uzunlugu (Kolon merkezlerinden)
Global KolDist=SisUz/(KolonSayisi.value-1)--Kolonlar arasi mesafe (merkezlerinden)
Global KolSay=KolonSayisi.value--Sistemin tek siradaki kolon sayisi
Global KolEni=SisGen/50--Kolon eni
Global KolKal=KolDist/40--Kolon kalinligi
Global KolYuks=KolonYuks.value--Kolon yüksekligi
Global KirYuks=SisGen/16--Kiris yüksekligi
Global KirGen=SisGen-KolEni--Kiris genisligi
Global KirAltH=KolYuks-KirYuks--Kirisin yerden yüksekligi
Global KirHoPrYuk=SisGen/200--Kiris yatay profil yükseklikleri(kalinliklari)
Global KirVePrEni=KirHoPrYuk/2--Kiris düsey profil enleri(kalinliklari)
Global KirVePrYuk=KirYuks-(2*KirHoPrYuk)--Kiris düsey profil yükseklikleri
Global VerPrSayKS=(Pow (Pow SisGen 2) 0.2)/(Pow (Pow 1000 2) 0.2)--Kiris düsey profil sayisi KatSayisi (Sistem genisligi disinda, profillerin sikligini arttirip azaltmak için kullanilan bir parametre..)
Global VerPrSay=((((KirGen/2)-(KirHoPrYuk+(KirVePrEni/2)))/KirVePrYuk)*VerPrSayKS)-1--Kiris düsey profil sayisi nin yarisi
Global VerPrSay = (VerPrSay as integer)--Sayiyi tam sayi yaptik...
Global VerPrDist=((KirGen/2)-(KirHoPrYuk+(KirVePrEni/2))-(VerPrSay*KirVePrEni))/(VerPrSay+1)--Kiris düsey profiller arasi mesafe
Global DiagPrUz=sqrt ((Pow KirVePrYuk 2) + (Pow VerPrDist 2))--Kiris diagonal profil uzunlugu
Global DiagPrXPo=(DiagPrUz*(KirVePrEni/2))/KirVePrYuk--Kiris diagonal profil köseden X yönündeki mesafesi
Global DiagPrYpo=KirVePrYuk*DiagPrXPo/VerPrDist--Kiris diagonal profil köseden Y yönündeki mesafesi
Global theModi = extrude amount:(KolKal/4)
if (KolDist) > 1000 or (KolDist) < 300 then messagebox "DIKKAT!!! Girilen parametler uygun degil. Parametrelerinizi gözden geçiriniz... (Kolonlar arasi mesafe 300cm ile 1000cm arasinda olmali)"
else
(
if (KolYuks-KirYuks) < 300 then messagebox "DIKKAT!!! Girilen parametler uygun degildir. Parametrelerinizi gözden geçiriniz... (Kiris yerden yüksekligi minumum 300cm olmali)"
else
(
(
CamDis = Targetcamera fov:40 nearclip:1 farclip:1000 nearrange:0 farrange:1000 mpassEnabled:off mpassRenderPerPass:off name:"CameraDis"
$CameraDis.target = Targetobject transform:(matrix3 [1,0,0] [0,1,0] [0,0,1] [SisGen/2, SisUz/10, KolYuks/2])
$CameraDis.pos = [4*SisGen/3, -(2*((13*SisGen/5))/3), 200]
CamIc = Targetcamera fov:65 nearclip:1 farclip:1000 nearrange:0 farrange:1000 mpassEnabled:off mpassRenderPerPass:off name:"CameraIc"
$CameraIc.target = Targetobject transform:(matrix3 [1,0,0] [0,1,0] [0,0,1] [SisGen/2, 4*SisUz/5, KolYuks/2])
$CameraIc.pos = [5*SisGen/6, -500, 200]
)
case RenderSistem.radbut_RenderType.state of --Render ve malzeme tipini seçer
(
1: (renderpresets.LoadAll 0 (SetFilePath + "Standard.rps")
--Steel1
meditMaterials[1] = Standardmaterial ()
meditMaterials[1].name = "Steel1"
meditMaterials[1].adLock = off
meditMaterials[1].ambient = color 80 80 80
meditMaterials[1].Diffuse = color 110 120 130
meditMaterials[1].specularLevel = 70
meditMaterials[1].glossiness = 50
--Steel2
meditMaterials[2] = Standardmaterial ()
meditMaterials[2].name = "Steel2"
meditMaterials[2].adLock = off
meditMaterials[2].ambient = color 80 80 80
meditMaterials[2].Diffuse = color 165 170 180
meditMaterials[2].specularLevel = 70
meditMaterials[2].glossiness = 50
--Roof
meditMaterials[7] = Standardmaterial ()
meditMaterials[7].name = "Roof"
meditMaterials[7].adLock = off
meditMaterials[7].ambient = color 80 80 80
meditMaterials[7].Diffuse = color 100 0 0
meditMaterials[7].specularLevel = 55
meditMaterials[7].glossiness = 30
--Panels
meditMaterials[8] = Standardmaterial ()
meditMaterials[8].name = "Panel"
meditMaterials[8].adLock = off
```

```
meditMaterials[8].ambient = color 80 80 80
meditMaterials[8].Diffuse = color 200 200 200
meditMaterials[8].specularLevel = 55
meditMaterials[8].glossiness = 30
--Ground
meditMaterials[13] = MatteShadow ()
meditMaterials[13].name = "Ground"
meditMaterials[13].opaqueAlpha = on
meditMaterials[13].applyAtmosphere = on
--BackGround için Vray
meditMaterials[14] = Standardmaterial ()
meditMaterials[14].name = "BackGroundVR"
meditMaterials[14].diffuseMap = Gradient ()
meditMaterials[14].diffuseMap.coordinates.mappingType = 1
meditMaterials[14].diffuseMap.coordinates.mapping = 3
meditMaterials[14].diffuseMap.name = "BG_Gradient"
meditMaterials[14].diffuseMap.color1 = color 0 72 255
meditMaterials[14].diffuseMap.color2 = color 201 223 223
meditMaterials[14].diffuseMap.color3 = color 255 254 237
meditMaterials[14].diffuseMap.map1 = Noise ()
meditMaterials[14].diffuseMap.map1.name = "BG-GR_Noise"
meditMaterials[14].diffuseMap.map1.color1 = color 0 62 154
meditMaterials[14].diffuseMap.map1.color2 = color 255 244 201
meditMaterials[14].diffuseMap.map1.size = 18
meditMaterials[14].diffuseMap.map1.phase = 7
--Environment Map ( BackGroundun diffuse unu kullaniyor)
environmentMap = Gradient ()
environmentMap = meditMaterials[14].diffuseMap
)
2: (renderpresets.LoadAll 0 (SetFilePath + "Vray.rps")
--Steel1
meditMaterials[1] = VRayMtl ()
meditMaterials[1].name = "Steel1"
meditMaterials[1].Diffuse = color 110 120 130
meditMaterials[1].Reflection = color 20 20 20
meditMaterials[1].reflection_lockGlossiness = off
meditMaterials[1].hilight_glossiness = 0.9
meditMaterials[1].reflection_glossiness = 0.9
--Steel2
meditMaterials[2] = VRayMtl ()
meditMaterials[2].name = "Steel2"
meditMaterials[2].Diffuse = color 165 170 180
meditMaterials[2].Reflection = color 20 20 20
meditMaterials[2].reflection_lockGlossiness = off
meditMaterials[2].hilight_glossiness = 0.9
meditMaterials[2].reflection_glossiness = 0.9
--Roof
meditMaterials[7] = VRayMtl ()
meditMaterials[7].name = "Roof"
meditMaterials[7].Diffuse = color 100 0 0
meditMaterials[7].Reflection = color 20 20 20
meditMaterials[7].reflection_lockGlossiness = off
meditMaterials[7].hilight_glossiness = 0.65
meditMaterials[7].reflection_glossiness = 0.9
--Panels
meditMaterials[8] = VRayMtl ()
meditMaterials[8].name = "Panel"
meditMaterials[8].Diffuse = color 200 200 200
meditMaterials[8].Reflection = color 20 20 20
meditMaterials[8].reflection_lockGlossiness = off
meditMaterials[8].hilight_glossiness = 0.65
meditMaterials[8].reflection_glossiness = 0.9
--Ground
meditMaterials[13] = VRayMtlWrapper ()
meditMaterials[13].name = "Ground"
meditMaterials[13].matteSurface = on
meditMaterials[13].matte_shadows = on
meditMaterials[13].matte_shadowsAffectAlpha = on
meditMaterials[13].baseMtl = VRayMtl ()
meditMaterials[13].baseMtl.name = "GroundVR"
meditMaterials[13].baseMtl.Diffuse = color 255 255 255
--BackGround için Vray
meditMaterials[14] = VRayMtl ()
meditMaterials[14].texmap_diffuse = Gradient ()
meditMaterials[14].texmap_diffuse.coordinates.mappingType = 1
meditMaterials[14].texmap_diffuse.coordinates.mapping = 3
meditMaterials[14].texmap_diffuse.color1 = color 0 72 255
meditMaterials[14].texmap_diffuse.color2 = color 201 223 223
meditMaterials[14].texmap_diffuse.color3 = color 255 254 237
meditMaterials[14].texmap_diffuse.noiseSize = 0
meditMaterials[14].texmap_diffuse.map1 = Noise ()
meditMaterials[14].texmap_diffuse.map1.color1 = color 0 62 154
meditMaterials[14].texmap_diffuse.map1.color2 = color 255 244 201
meditMaterials[14].texmap_diffuse.map1.size = 18
meditMaterials[14].texmap_diffuse.map1.phase = 7
meditMaterials[14].name = "BackGroundVR"
meditMaterials[14].texmap_diffuse.name = "BG_Gradient"
meditMaterials[14].texmap_diffuse.map1.name = "BG-GR_Noise"

--Environment Map ( BackGroundun diffuse unu kullaniyor)
environmentMap = Gradient ()
environmentMap = meditMaterials[14].texmap_diffuse
))
fn KirSist =
(
--Kolonlarin olusturulmasi
KolonSol = box width:KolEni length:KolKal height:KolYuks name: "KolonSol" wirecolor:gray
KolonSag = instance KolonSol name:"KolonSag" wirecolor:gray
Move KolonSag [SisGen, 0, 0]
--Kiris Olusturma
KirAlt = box width:KirGen length:(KolKal/2) height:KirHoPrYuk name: "KirisAltProfil" wirecolor:orange --Alt baslik
Move KirAlt [(SisGen/2), 0, KirAltH]
KirUst = instance KirAlt name: "KirisUstProfil" wirecolor:orange --Üst baslik
move KirUst [0, 0, (KirYuks-KirHoPrYuk)]
KirPrSol = box width:KirHoPrYuk length:(KolKal/2) height:KirVePrYuk name: "KirisSolProfil" wirecolor:orange --Baslik sol profil
Move KirPrSol [(KolEni/2)+(KirHoPrYuk/2), 0, KirAltH+KirHoPrYuk]
KirPrSag = instance KirPrSol name: "KirisSagProfil" wirecolor:orange --Baslik sag profil
Move KirPrSag [KirGen-KirHoPrYuk, 0, 0]
for i = 0 to (2*VerPrSay) do
(
KirPrVer = box width:KirVePrEni length:(KolKal/4) height:KirVePrYuk name: "KirisVerticalProfil" wirecolor:white --Vertical profiller
Move KirPrVer [((KolEni/2)+KirHoPrYuk+(KirVePrEni/2)+VerprDist), 0, KirAltH+KirHoPrYuk]
Move KirPrVer [(VerprDist+KirVePrEni)*i,0,0]
)
for m = 0 to VerPrSay do
(
DiagPrA = SplineShape pos:[0,0,0] name: "KirisDiagonalProfilA" --Diagonal Profiller
addNewSpline DiagPrA
addKnot DiagPrA 1 #corner #line [0,0,0]
addKnot DiagPrA 1 #corner #line [DiagPrXPo,0,0]
addKnot DiagPrA 1 #corner #line [VerPrDist,(KirVePrYuk-DiagPrYPo),0]
addKnot DiagPrA 1 #corner #line [VerPrDist,KirVePrYuk,0]
addKnot DiagPrA 1 #corner #line [(VerPrDist-DiagPrXPo),KirVePrYuk,0]
addKnot DiagPrA 1 #corner #line [0,DiagPrYPo,0]
Close DiagPrA 1
DiagPrA.wirecolor = color 255 255 255
rotate DiagPrA (eulerangles 90 0 0)
updateShape DiagPrA
Move DiagPrA [(KolEni/2)+KirHoPrYuk+((VerprDist+KirVePrEni)*(VerPrSay+1)), (KolKal/8), KirAltH+KirHoPrYuk]
--Move DiagPrA [(KolEni/2)+KirHoPrYuk, (KolKal/8), KirAltH+KirHoPrYuk]
Move DiagPrA [(VerprDist+KirVePrEni)*m,0,0]
addmodifier DiagPrA theModi
)
for n = 0 to VerPrSay do
(
DiagPrB = SplineShape pos:[0,0,0] name: "KirisDiagonalProfilB" --Diagonal Profiller
addNewSpline DiagPrB
addKnot DiagPrB 1 #corner #line [0,0,0]
addKnot DiagPrB 1 #corner #line [-DiagPrXPo,0,0]
addKnot DiagPrB 1 #corner #line [-VerPrDist,(KirVePrYuk-DiagPrYPo),0]
addKnot DiagPrB 1 #corner #line [-VerPrDist,KirVePrYuk,0]
addKnot DiagPrB 1 #corner #line [-(VerPrDist-DiagPrXPo),KirVePrYuk,0]
addKnot DiagPrB 1 #corner #line [0,DiagPrYPo,0]
Close DiagPrB 1
DiagPrB.wirecolor = color 255 255 255
rotate DiagPrB (eulerangles 90 0 0)
updateShape DiagPrB
Move DiagPrB [(KolEni/2)+KirHoPrYuk+VerPrDist, (KolKal/8), KirAltH+KirHoPrYuk]
--Move DiagPrB [(KolEni/2)+KirHoPrYuk+((VerprDist+KirVePrEni)*(VerPrSay+2))-KirVePrEni, (KolKal/8), KirAltH+KirHoPrYuk]
Move DiagPrB [(VerprDist+KirVePrEni)*n,0,0]
addmodifier DiagPrB theModi
)) --Fn KirSist Kapama
for t = 0 to KolSay-1 do--Kiris-Kolon sistemimizi burada tamamliyoruz ve tüm akslara kopyaliyoruz (instance olarak)
(
KirSist()
$kirisVerticalProfil*.baseobject = $kirisVerticalProfil*[1].baseobject
$KirisDiagonalProfilA*.baseobject = $KirisDiagonalProfilA*[1].baseobject
$KirisDiagonalProfilB*.baseobject = $KirisDiagonalProfilB*[1].baseobject
$KolonSol*.baseobject = $KolonSol*[1].baseobject
$KolonSag*.baseobject = $KolonSag*[1].baseobject
$KirisAltProfil*.baseobject = $KirisAltProfil*[1].baseobject
$KirisUstProfil*.baseobject = $KirisUstProfil*[1].baseobject
$KirisSolProfil*.baseobject = $KirisSolProfil*[1].baseobject
$KirisSagProfil*.baseobject = $KirisSagProfil*[1].baseobject
$KolonS*.material = meditMaterials[1]
$Kiris*.material = meditMaterials[2]
KirisSistem = group geometry name:"KirisSistem" select:false
Move KirisSistem [0, KolDist*t, 0]
)
for v = 0 to KolSay-2 do--Kiris sistemleri arasindaki düz profil krisleri yerlestirdik (yine instance olarak)
(
KirDuzSol = box width:(KolEni/2) length:(KolDist-KolKal) height:(KolDist/16) pos:[(-KolEni/4), (KolDist/2), (KolYuks-(KolDist/16))] name: "KirisDuzSol" Wirecolor:white
KirDuzSag = instance KirDuzSol name: "KirisDuzSag" Wirecolor:white
```

```
Move KirDuzSag [(Sisgen+(KolEni/2)), 0, 0]
Move KirDuzSol [0,KolDist*v,0]
Move KirDuzSag [0,KolDist*v,0]
$KirisDuzSol*.baseobject = $KirisDuzSol*[1].baseobject
$KirisDuzSag*.baseobject = $KirisDuzSag*[1].baseobject
$KirisDuz*.material = meditMaterials[2]
)
if SistemGorunus.chbox_duvar.checked then
(
for y = 0 to (2*KolSay)-5 do--Yan paneller - Ortadakiler (instance olarak)
(
PanelSol = box width:10 length:((KolDist/2)-2) height:KolYuks pos:[(-KolEni/2)-5, (KolDist/2)+(KolDist/4), 0] name: "PanelSol"  Wirecolor:(color 100 150 200)
PanelSag = instance PanelSol name: "PanelSag"  Wirecolor:(color 100 150 200)
Move PanelSag [(Sisgen+KolEni+10), 0, 0]
Move PanelSol [0,(KolDist/2)*y,0]
Move PanelSag [0,(KolDist/2)*y,0]
$PanelSol*.baseobject = $PanelSol*[1].baseobject
$PanelSag*.baseobject = $PanelSag*[1].baseobject
$Panel*.material = meditMaterials[8]
)
for z = 0 to 1 do--Yan paneller - Bastakiler (instance olarak)
(
PanelSol1 = box width:10 length:((KolDist/2)-2+(KolKal/2)+1) height:KolYuks pos:[(-KolEni/2)-5, (KolDist/4)-(KolKal/4)-0.5, 0] name: "PanelSol1" Wirecolor:(color 100 150 200)
PanelSag1 = instance PanelSol1 name: "PanelSag1"  Wirecolor:(color 100 150 200)
Move PanelSag1 [(Sisgen+KolEni+10), 0, 0]
Move PanelSol1 [0,(SisUz-(KolDist/2)+8.5)*z,0]
Move PanelSag1 [0,(SisUz-(KolDist/2)+8.5)*z,0]
$PanelSol1*.baseobject = $PanelSol1*[1].baseobject
$PanelSag1*.baseobject = $PanelSag1*[1].baseobject
$Panel*.material = meditMaterials[8]
))
Zemin = plane width:(SisGen*3) length:(SisUz*3) pos:[(SisGen/2), (SisUz/2), 0] name:"Zemin" wirecolor:White  --Zemin olusturur
$Zemin*.material = meditMaterials[13]
if SistemGorunus.chbox_cati.checked then  --Çatiyi olusturur duruma göre
Cati = box width:(SisGen+KolEni+20) length:(SisUz+KolKal) height:20 pos:[(SisGen/2), (SisUz/2), KolYuks] name: "Cati" wirecolor:(color 150 0 0)
$Cati*.material = meditMaterials[7]
case RenderSistem.radbut_Cam_type.state of  --Kameralari olusturur ve seçimini yapar
(
1: (select $CameraDis)
2: (select $CameraIc)
)
max vpt camera
deselect objects
backgroundColor = color 255 255 255
if RenderSistem.chbox_render.checked and (RenderSistem.radbut_render_time.state == 2) then  --Render seçimini yapar ve render alir
max quick render
if (RenderSistem.radbut_render_time.state == 1) then  --Render alir ama önce sorar
if queryBox "Sonra hatirlat demistin. Render alayim mi?" beep:false then
max quick render
)  --If-Else kapama
)  --If-Else kapama
)  --On-do kapama
on buttGeri pressed do
(
addRollout bilgiParalel PSS
addRollout ParalelKafesKiris PSS
removeRollout bilgiParalelPratt PSS
removeRollout SisParaPlPratt PSS
removeRollout SistemGorunus PSS
removeRollout RenderSistem PSS
removeRollout Exporting PSS
))  --Rollout kapama
--addRollout bilgiParalelPratt PSS
--addRollout SisParaPlPratt PSS
--addRollout SistemGorunus PSS
--addRollout RenderSistem PSS
```

ParalelPRATT_Modified.ms

```
--PSS = newRolloutFloater "P.S.S. 1.01 Beta" 240 770 800 100
GetFilePath = getThisScriptFilename()
SetFilePath= getFileNamePath GetFilePath
rollout bilgiParalelPrattM "Kisa Bilgi"  RolledUp:true
(
group "Script Kullanimi"
(
label lab1 "--Asagidaki paralel baslikli PRATT-Modified" align:#left
label lab2 "      kafes kirisi sistem özelliklerinizi seçiniz" align:#left
label lab3 "      ve render - malzeme ayarlarini yapiniz." align:#left
))
 rollout SisParaPlPrattM "Sistem Parametreleri"
(
button buttGeri "<<< Üst Menüye Geri Dön <<<" border:true width:220 height:30 align:#center toolTip:"Bir Önceki Menüye Geri Döner"
group "Açikliklar"
(
spinner SistemGenis "Sistem Genisligi (cm):" range:[500,5000,2000] type:#float toolTip:"Sistemin Geçecegi Mesafeyi Belirler. (5m-50m)"
spinner SistemUzun "Sistem Uzunlugu (cm):" range:[500,30000,3000] type:#float  toolTip:"Yapinin Toplam Uzunlugunu Belirler. (5m-300m)"
)
group "Kolonlar"
(
spinner KolonYuks "Kolon Yüksekligi (cm):" range:[400,2000,800] type:#float toolTip:"Sistemin Yerden En Yüksek Noktasina Olan Mesafeyi Belirler. (4m-20m)"
spinner KolonSayisi "Kolon Sayisi:" range:[2,33,6] type:#integer toolTip:"Sistem Uzunlugu Yönündeki Toplam Aks Sayisini Belirler. (2-33)"
)
group "Diger Anlik Parametrik Bilgiler"
(
label labX1 "Kiris Yüksekligi: 125.0 cm" align:#left
label labX2 "Kirisin Yerden Yüksekligi: 675.0 cm" align:#left
label labX3 "Kolonlar Arasi Mesafe: 600.0 cm" align:#left
)
fn Paraminfo =
(
labX1.text = "Kiris Yüksekligi: " + (SistemGenis.value/16) as string + " cm"
labX2.text = "Kirisin Yerden Yüksekligi: " + (KolonYuks.value-(SistemGenis.value/16)) as string + " cm"
labX3.text = "Kolonlar Arasi Mesafe: " + ((SistemUzun.value)/(KolonSayisi.value-1)) as string + " cm"
)
on SistemGenis changed value do  Paraminfo()
on SistemUzun changed value do Paraminfo()
on KolonYuks changed value do Paraminfo()
on KolonSayisi changed value do Paraminfo()
button sil "Sil" border:true width:135 height:20 align:#left offset:[-10,0] across:2 toolTip:"Sahnedeki Bütün Nesneleri Siler."
on sil pressed do delete objects
button resett "Reset" border:true width:80 height:20 align:#right offset:[10,0] toolTip:"Sahneye Reset Atar (Yeniden Baslatir)."
on resett pressed do resetMaxFile()
button Olustur "Olustur" border:true width:220 height:30 align:#center offset:[0,0] toolTip:"Belirlediginiz Parametreler Dahilinde Sisteminizi Olusturur."
on Olustur pressed do
(
Global SisGen=SistemGenis.value--Sistem genisligi (Kolon merkezlerinden)
Global SisUz=SistemUzun.value--Sistem uzunlugu (Kolon merkezlerinden)
Global KolDist=SisUz/(KolonSayisi.value-1)--Kolonlar arasi mesafe (merkezlerinden)
Global KolSay=KolonSayisi.value--Sistemin tek siradaki kolon sayisi
Global KolEni=SisGen/50--Kolon eni
Global KolKal=KolDist/40--Kolon kalinligi
Global KolYuks=KolonYuks.value--Kolon yüksekligi
Global KirYuks=SisGen/16--Kiris yüksekligi
Global KirGen=SisGen+KolEni--Kiris genisligi
Global KirAltH=KolYuks-KirYuks--Kirisin yerden yüksekligi
Global KirHoPrYuk=SisGen/200--Kiris yatay profil yükseklikleri(kalinliklari)
Global KirVePrEni=KirHoPrYuk/2--Kiris düsey profil enleri(kalinliklari)
Global KirVePrYuk=KirYuks-(2*KirHoPrYuk)--Kiris düsey profil yükseklikleri
Global VerPrSayKS=(Pow (Pow SisGen 2) 0.2)/(Pow (Pow 1000 2) 0.2)--Kiris düsey profil sayisi KatSayisi (Sistem genisligi disinda, profillerin sikligini arttirip azaltmak için kullanilan bir parametre..)
Global VerPrSay=((((KirGen/2)-(KirVePrEni/2))/KirVePrYuk)*VerPrSayKS)-1--Kiris düsey profil sayisi nin yarisi
Global VerPrSay = (VerPrSay as integer)--Sayiyi tam sayi yaptik...
Global VerPrDist=((KirGen/2)-(KirVePrEni/2)-(VerPrSay*KirVePrEni))/(VerPrSay+1)--Kiris düsey profiller arasi mesafe
Global DiagPrUz=sqrt ((Pow KirVePrYuk 2) + (Pow VerPrDist 2))--Kiris diagonal profil uzunlugu
Global DiagPrXPo=(DiagPrUz*(KirVePrEni/2))/KirVePrYuk--Kiris diagonal profil köseden X yönündeki mesafesi
Global DiagPrYpo=KirVePrYuk*DiagPrXPo/VerPrDist--Kiris diagonal profil köseden Y yönündeki mesafesi
Global theModi = extrude amount:(KolKal/4)
Global theModi2 = extrude amount:(KolKal/2)
Global DiagPr1XPo=VerPrDist-((VerPrDist*KirVePrYuk)/(KirHoPrYuk+KirVePrYuk))--Kiris diagonal profil köseden X yönündeki mesafesi
if (KolDist) > 1000 or (KolDist) < 300 then messagebox "DIKKAT!!! Girilen parametler uygun degil. Parametrelerinizi gözden geçiriniz... (Kolonlar arasi mesafe 300cm ile 1000cm arasinda olmali)"
else
(
if (KolYuks-KirYuks) < 300 then messagebox "DIKKAT!!! Girilen parametler uygun degildir. Parametrelerinizi gözden geçiriniz... (Kiris yerden yüksekligi minumum 300cm olmali)"
else
((
CamDis = Targetcamera fov:40 nearclip:1 farclip:1000 nearrange:0 farrange:1000 mpassEnabled:off mpassRenderPerPass:off name:"CameraDis"
$CameraDis.target = Targetobject transform:(matrix3 [1,0,0] [0,1,0] [0,0,1] [SisGen/2, SisUz/10, KolYuks/2])
$CameraDis.pos = [4*SisGen/3, -(2*((13*SisGen/5))/3), 200]
CamIc = Targetcamera fov:65 nearclip:1 farclip:1000 nearrange:0 farrange:1000 mpassEnabled:off mpassRenderPerPass:off name:"CameraIc"
$CameraIc.target = Targetobject transform:(matrix3 [1,0,0] [0,1,0] [0,0,1] [SisGen/2, 4*SisUz/5, KolYuks/2])
$CameraIc.pos = [5*SisGen/6, -500, 200]
)
```

```
case RenderSistem.radbut_RenderType.state of  --Render ve malzeme tipini seçer
(
1: (renderpresets.LoadAll 0 (SetFilePath + "Standard.rps")
--Steel1
meditMaterials[1] = Standardmaterial ()
meditMaterials[1].name = "Steel1"
meditMaterials[1].adLock = off
meditMaterials[1].ambient = color 80 80 80
meditMaterials[1].Diffuse = color 110 120 130
meditMaterials[1].specularLevel = 70
meditMaterials[1].glossiness = 50
--Steel2
meditMaterials[2] = Standardmaterial ()
meditMaterials[2].name = "Steel2"
meditMaterials[2].adLock = off
meditMaterials[2].ambient = color 80 80 80
meditMaterials[2].Diffuse = color 165 170 180
meditMaterials[2].specularLevel = 70
meditMaterials[2].glossiness = 50
--Roof
meditMaterials[7] = Standardmaterial ()
meditMaterials[7].name = "Roof"
meditMaterials[7].adLock = off
meditMaterials[7].ambient = color 80 80 80
meditMaterials[7].Diffuse = color 100 0 0
meditMaterials[7].specularLevel = 55
meditMaterials[7].glossiness = 30
--Panels
meditMaterials[8] = Standardmaterial ()
meditMaterials[8].name = "Panel"
meditMaterials[8].adLock = off
meditMaterials[8].ambient = color 80 80 80
meditMaterials[8].Diffuse = color 200 200 200
meditMaterials[8].specularLevel = 55
meditMaterials[8].glossiness = 30
--Ground
meditMaterials[13] = MatteShadow ()
meditMaterials[13].name = "Ground"
meditMaterials[13].opaqueAlpha = on
meditMaterials[13].applyAtmosphere = on
--BackGround için Vray
meditMaterials[14] = Standardmaterial ()
meditMaterials[14].name = "BackGroundVR"
meditMaterials[14].diffuseMap = Gradient ()
meditMaterials[14].diffuseMap.coordinates.mappingType = 1
meditMaterials[14].diffuseMap.coordinates.mapping = 3
meditMaterials[14].diffuseMap.name = "BG_Gradient"
meditMaterials[14].diffuseMap.color1 = color 0 72 255
meditMaterials[14].diffuseMap.color2 = color 201 223 223
meditMaterials[14].diffuseMap.color3 = color 255 254 237
meditMaterials[14].diffuseMap.map1 = Noise ()
meditMaterials[14].diffuseMap.map1.name = "BG-GR_Noise"
meditMaterials[14].diffuseMap.map1.color1 = color 0 62 154
meditMaterials[14].diffuseMap.map1.color2 = color 255 244 201
meditMaterials[14].diffuseMap.map1.size = 18
meditMaterials[14].diffuseMap.map1.phase = 7
--Environment Map ( BackGroundun diffuse unu kullaniyor)
environmentMap = Gradient ()
environmentMap = meditMaterials[14].diffuseMap
)
2: (renderpresets.LoadAll 0 (SetFilePath + "Vray.rps")
--Steel1
meditMaterials[1] = VRayMtl ()
meditMaterials[1].name = "Steel1"
meditMaterials[1].Diffuse = color 110 120 130
meditMaterials[1].Reflection = color 20 20 20
meditMaterials[1].reflection_lockGlossiness = off
meditMaterials[1].hilight_glossiness = 0.9
meditMaterials[1].reflection_glossiness = 0.9
--Steel2
meditMaterials[2] = VRayMtl ()
meditMaterials[2].name = "Steel2"
meditMaterials[2].Diffuse = color 165 170 180
meditMaterials[2].Reflection = color 20 20 20
meditMaterials[2].reflection_lockGlossiness = off
meditMaterials[2].hilight_glossiness = 0.9
meditMaterials[2].reflection_glossiness = 0.9
--Roof
meditMaterials[7] = VRayMtl ()
meditMaterials[7].name = "Roof"
meditMaterials[7].Diffuse = color 100 0 0
meditMaterials[7].Reflection = color 20 20 20
meditMaterials[7].reflection_lockGlossiness = off
meditMaterials[7].hilight_glossiness = 0.65
meditMaterials[7].reflection_glossiness = 0.9
--Panels
meditMaterials[8] = VRayMtl ()
meditMaterials[8].name = "Panel"
meditMaterials[8].Diffuse = color 200 200 200
meditMaterials[8].Reflection = color 20 20 20
meditMaterials[8].reflection_lockGlossiness = off
meditMaterials[8].hilight_glossiness = 0.65
meditMaterials[8].reflection_glossiness = 0.9
--Ground
meditMaterials[13] = VRayMtlWrapper ()
meditMaterials[13].name = "Ground"
meditMaterials[13].matteSurface = on
meditMaterials[13].matte_shadows = on
meditMaterials[13].matte_shadowsAffectAlpha = on
meditMaterials[13].baseMtl = VRayMtl ()
meditMaterials[13].baseMtl.name = "GroundVR"
meditMaterials[13].baseMtl.Diffuse = color 255 255 255
--BackGround için Vray
meditMaterials[14] = VRayMtl ()
meditMaterials[14].texmap_diffuse = Gradient ()
meditMaterials[14].texmap_diffuse.coordinates.mappingType = 1
meditMaterials[14].texmap_diffuse.coordinates.mapping = 3
meditMaterials[14].texmap_diffuse.color1 = color 0 72 255
meditMaterials[14].texmap_diffuse.color2 = color 201 223 223
meditMaterials[14].texmap_diffuse.color3 = color 255 254 237
meditMaterials[14].texmap_diffuse.noiseSize = 0
meditMaterials[14].texmap_diffuse.map1 = Noise ()
meditMaterials[14].texmap_diffuse.map1.color1 = color 0 62 154
meditMaterials[14].texmap_diffuse.map1.color2 = color 255 244 201
meditMaterials[14].texmap_diffuse.map1.size = 18
meditMaterials[14].texmap_diffuse.map1.phase = 7
meditMaterials[14].name = "BackGroundVR"
meditMaterials[14].texmap_diffuse.name = "BG_Gradient"
meditMaterials[14].texmap_diffuse.map1.name = "BG-GR_Noise"
--Environment Map ( BackGroundun diffuse unu kullaniyor)
environmentMap = Gradient ()
environmentMap = meditMaterials[14].texmap_diffuse
))
fn KirSist =
(
--Kolonlarin olusturulmasi
KolonSol = box width:KolEni length:KolKal height:KirAltH name: "KolonSol" wirecolor:gray
KolonSag = instance KolonSol name:"KolonSag" wirecolor:gray
Move KolonSag [SisGen, 0, 0]
--Kiris Olusturma
KirAlt = box width:KirGen length:(KolKal/2) height:KirHoPrYuk name: "KirisAltProfil" wirecolor:orange --Alt baslik
Move KirAlt [(SisGen/2), 0, KirAltH]
KirUst = copy KirAlt name: "KirisUstProfil" wirecolor:orange --Üst baslik
KirUst.width = (KirGen-(2*VerPrDist))
move KirUst [0, 0, (KirYuks-KirHoPrYuk)]
for i = 0 to (2*VerPrSay) do
(
KirPrVer = box width:KirVePrEni length:(KolKal/4) height:KirVePrYuk name: "KirisVerticalProfil" wirecolor:white --Vertical profiller
Move KirPrVer [((KirVePrEni/2)+VerprDist-(KolEni/2)), 0, KirAltH+KirHoPrYuk]
Move KirPrVer [(VerprDist+KirVePrEni)*i,0,0]
)
DiagPrSol = SplineShape pos:[0,0,0] name: "KirisDiagonalProfilSolBas" --Diagonal Profil Sol Bastaki
addNewSpline DiagPrSol
addKnot DiagPrSol 1 #corner #line [0,0,0]
addKnot DiagPrSol 1 #corner #line [DiagPr1XPo,0,0]
addKnot DiagPrSol 1 #corner #line [VerPrDist,KirVePrYuk,0]
addKnot DiagPrSol 1 #corner #line [VerPrDist,KirVePrYuk+KirHoPrYuk,0]
Close DiagPrSol 1
DiagPrSol.wirecolor = orange
rotate DiagPrSol (eulerangles 90 0 0)
updateShape DiagPrSol
Move DiagPrSol [-(KolEni/2), (KolKal/4), KirAltH+KirHoPrYuk]
addmodifier DiagPrSol theModi2
DiagPrSag = SplineShape pos:[0,0,0] name: "KirisDiagonalProfilSagBas" --Diagonal Profil Sag Bastaki
addNewSpline DiagPrSag
addKnot DiagPrSag 1 #corner #line [0,0,0]
addKnot DiagPrSag 1 #corner #line [-DiagPr1XPo,0,0]
addKnot DiagPrSag 1 #corner #line [-VerPrDist,KirVePrYuk,0]
addKnot DiagPrSag 1 #corner #line [-VerPrDist,KirVePrYuk+KirHoPrYuk,0]
Close DiagPrSag 1
DiagPrSag.wirecolor = orange
rotate DiagPrSag (eulerangles 90 0 0)
updateShape DiagPrSag
Move DiagPrSag [KirGen-(KolEni/2), (KolKal/4), KirAltH+KirHoPrYuk]
addmodifier DiagPrSag theModi2
for m = 0 to VerPrSay-1 do
(
DiagPrA = SplineShape pos:[0,0,0] name: "KirisDiagonalProfilA"  --Diagonal Profiller
addNewSpline DiagPrA
addKnot DiagPrA 1 #corner #line [0,0,0]
addKnot DiagPrA 1 #corner #line [DiagPrXPo,0,0]
addKnot DiagPrA 1 #corner #line [VerPrDist,(KirVePrYuk-DiagPrYPo),0]
addKnot DiagPrA 1 #corner #line [VerPrDist,KirVePrYuk,0]
addKnot DiagPrA 1 #corner #line [(VerPrDist-DiagPrXPo),KirVePrYuk,0]
addKnot DiagPrA 1 #corner #line [0,DiagPrYPo,0]
Close DiagPrA 1
DiagPrA.wirecolor = color 255 255 255
rotate DiagPrA (eulerangles 90 0 0)
updateShape DiagPrA
Move DiagPrA [-(KolEni/2)+((VerPrDist+KirVePrEni)*(VerPrSay+2))-KirVePrEni-VerPrDist, (KolKal/8), KirAltH+KirHoPrYuk]
```

```
Move DiagPrA [(VerPrDist+KirVePrEni)*m,0,0]
addmodifier DiagPrA theModi
)
for n = 1 to VerPrSay do
(
DiagPrB = SplineShape pos:[0,0,0] name: "KirisDiagonalProfilB"  --Diagonal Profiller
addNewSpline DiagPrB
addKnot DiagPrB 1 #corner #line [0,0,0]
addKnot DiagPrB 1 #corner #line [-DiagPrXPo,0,0]
addKnot DiagPrB 1 #corner #line [-VerPrDist,(KirVePrYuk-DiagPrYPo),0]
addKnot DiagPrB 1 #corner #line [-VerPrDist,KirVePrYuk,0]
addKnot DiagPrB 1 #corner #line [-(VerPrDist-DiagPrXPo),KirVePrYuk,0]
addKnot DiagPrB 1 #corner #line [0,DiagPrYPo,0]
Close DiagPrB 1
DiagPrB.wirecolor = color 255 255 255
rotate DiagPrB (eulerangles 90 0 0)
updateShape DiagPrB
Move DiagPrB [-(KolEni/2)+VerPrDist, (KolKal/8), KirAltH+KirHoPrYuk]
Move DiagPrB [(VerPrDist+KirVePrEni)*n,0,0]
addmodifier DiagPrB theModi
))  --Fn KirSist Kapama
for t = 0 to KolSay-1 do--Kiris-Kolon sistemimizi burada tamamliyoruz ve tüm akslara kopyaliyoruz (instance olarak)
(
KirSist()
$kirisVerticalProfil*.baseobject = $kirisVerticalProfil*[1].baseobject
$KirisDiagonalProfilA*.baseobject = $KirisDiagonalProfilA*[1].baseobject
$KirisDiagonalProfilB*.baseobject = $KirisDiagonalProfilB*[1].baseobject
$KolonSol*.baseobject = $KolonSol*[1].baseobject
$KolonSag*.baseobject = $KolonSag*[1].baseobject
$KolonS*.material = meditMaterials[1]
$Kiris*.material = meditMaterials[2]
KirisSistem = group geometry name:"KirisSistem" select:false
Move KirisSistem [0, KolDist*t, 0]
)
for v = 0 to KolSay-2 do--Kiris sistemleri arasindaki düz profil krisleri yerlestirdik (yine instance olarak)
(
KirDuzSol = box width:(KolEni/2) length:(KolDist-KolKal) height:(KolDist/16) pos:[(-KolEni/4), (KolDist/2), (KirAltH-(KolDist/16))] name: "KirisDuzSol" Wirecolor:white
KirDuzSag = instance KirDuzSol name: "KirisDuzSag" Wirecolor:white
Move KirDuzSag [(Sisgen+(KolEni/2)), 0, 0]
Move KirDuzSol [0,KolDist*v,0]
Move KirDuzSag [0,KolDist*v,0]
$KirisDuzSol*.baseobject = $KirisDuzSol*[1].baseobject
$KirisDuzSag*.baseobject = $KirisDuzSag*[1].baseobject
$KirisDuz*.material = meditMaterials[2]
)
if SistemGorunus.chbox_duvar.checked then
(
for y = 0 to (2*KolSay)-5 do--Yan paneller - Ortadakiler (instance olarak)
(
PanelSol = box width:10 length:((KolDist/2)-2) height:KirAltH pos:[(-KolEni/2)-5, (KolDist/2)+(KolDist/4), 0] name: "PanelSol"  Wirecolor:(color 100 150 200)
PanelSag = instance PanelSol name: "PanelSag"  Wirecolor:(color 100 150 200)
Move PanelSag [(Sisgen+KolEni+10), 0, 0]
Move PanelSol [0,(KolDist/2)*y,0]
Move PanelSag [0,(KolDist/2)*y,0]
$PanelSol*.baseobject = $PanelSol*[1].baseobject
$PanelSag*.baseobject = $PanelSag*[1].baseobject
$Panel*.material = meditMaterials[8]
)
for z = 0 to 1 do--Yan paneller - Bastakiler (instance olarak)
(
PanelSol1 = box width:10 length:((KolDist/2)-2+(KolKal/2)+1) height:KirAltH pos:[(-KolEni/2)-5, (KolDist/4)-(KolKal/4)-0.5, 0] name: "PanelSol1" Wirecolor:(color 100 150 200)
PanelSag1 = instance PanelSol1 name: "PanelSag1"  Wirecolor:(color 100 150 200)
Move PanelSag1 [(Sisgen+KolEni+10), 0, 0]
Move PanelSol1 [0,(SisUz-(KolDist/2)+8.5)*z,0]
Move PanelSag1 [0,(SisUz-(KolDist/2)+8.5)*z,0]
$PanelSol1*.baseobject = $PanelSol1*[1].baseobject
$PanelSag1*.baseobject = $PanelSag1*[1].baseobject
$Panel*.material = meditMaterials[8]
))
Zemin = plane width:(SisGen*3) length:(SisUz*3) pos:[(SisGen/2), (SisUz/2), 0] name:"Zemin" wirecolor:White  --Zemin olusturur
$Zemin*.material = meditMaterials[13]
if SistemGorunus.chbox_cati.checked then  --Çatiyi olusturur duruma göre
Cati = box width:(SisGen+KolEni-(2*VerPrDist)) length:(SisUz+KolKal) height:20 pos:[(SisGen/2), (SisUz/2), KolYuks] name: "Cati" wirecolor:(color 150 0 0)
$Cati*.material = meditMaterials[7]
case RenderSistem.radbut_Cam_type.state of  --Kameralari olusturur ve seçimini yapar
(
1: (select $CameraDis)
2: (select $CameraIc)
)
max vpt camera
deselect objects
backgroundColor = color 255 255 255
if RenderSistem.chbox_render.checked and (RenderSistem.radbut_render_time.state == 2) then  --Render seçimini yapar ve render alir
max quick render
if (RenderSistem.radbut_render_time.state == 1) then  --Render alir ama önce sorar
if queryBox "Sonra hatirlat demistin. Render alayim mi?" beep:false then
max quick render
)  --If-Else kapama
)  --If-Else kapama
)  --On-do kapama
on buttGeri pressed do
(
addRollout bilgiParalel PSS
addRollout ParalelKafesKiris PSS
removeRollout bilgiParalelPrattM PSS
removeRollout SisParaPlPrattM PSS
removeRollout SistemGorunus PSS
removeRollout RenderSistem PSS
removeRollout Exporting PSS
))  --Rollout kapama
--addRollout bilgiParalelPrattM PSS
--addRollout SisParaPlPrattM PSS
--addRollout SistemGorunus PSS
--addRollout RenderSistem PSS
```

ParalelWARREN.ms

```
--PSS = newRolloutFloater "P.S.S. 1.01 Beta" 240 770 800 100
GetFilePath = getThisScriptFilename()
SetFilePath= getFileNamePath GetFilePath
rollout bilgiParalelWarren "Kisa Bilgi"  RolledUp:true
(
group "Script Kullanimi"
(
label lab1 "--Asagidaki paralel baslikli WARREN kafes" align:#left
label lab2 "      kirisi sistem özelliklerinizi seçiniz ve" align:#left
label lab3 "      render - malzeme ayarlarini yapiniz." align:#left
)) rollout SisParaPlWarren "Sistem Parametreleri"
(
button buttGeri "<<< Üst Menüye Geri Dön <<<" border:true width:220 height:30 align:#center toolTip:"Bir Önceki Menüye Geri Döner"
group "Açikliklar"
(
spinner SistemGenis "Sistem Genisligi (cm):" range:[500,5000,2000] type:#float toolTip:"Sistemin Geçecegi Mesafeyi Belirler. (5m-50m)"
spinner SistemUzun "Sistem Uzunlugu (cm):" range:[500,30000,3000] type:#float  toolTip:"Yapinin Toplam Uzunlugunu Belirler. (5m-300m)"
)
group "Kolonlar"
(
spinner KolonYuks "Kolon Yüksekligi (cm):" range:[400,2000,800] type:#float toolTip:"Sistemin Yerden En Yüksek Noktasina Olan Mesafeyi Belirler. (4m-20m)"
spinner KolonSayisi "Kolon Sayisi:" range:[2,33,6] type:#integer toolTip:"Sistem Uzunlugu Yönündeki Toplam Aks Sayisini Belirler. (2-33)"
)
group "Diger Anlik Parametrik Bilgiler"
(
label labX1 "Kiris Yüksekligi: 125.0 cm" align:#left
label labX2 "Kirisin Yerden Yüksekligi: 675.0 cm" align:#left
label labX3 "Kolonlar Arasi Mesafe: 600.0 cm" align:#left
)
fn Paraminfo =
(
labX1.text = "Kiris Yüksekligi: " + (SistemGenis.value/16) as string + " cm"
labX2.text = "Kirisin Yerden Yüksekligi: " + (KolonYuks.value-(SistemGenis.value/16)) as string + " cm"
labX3.text = "Kolonlar Arasi Mesafe: " + ((SistemUzun.value)/(KolonSayisi.value-1)) as string + " cm"
)
on SistemGenis changed value do  Paraminfo()
on SistemUzun changed value do Paraminfo()
on KolonYuks changed value do Paraminfo()
on KolonSayisi changed value do Paraminfo()
button sil "Sil" border:true width:135 height:20 align:#left offset:[-10,0] across:2 toolTip:"Sahnedeki Bütün Nesneleri Siler."
on sil pressed do delete objects
button resett "Reset" border:true width:80 height:20 align:#right offset:[10,0] toolTip:"Sahneye Reset Atar (Yeniden Baslatir)."
on resett pressed do resetMaxFile()
button Olustur "Olustur" border:true width:220 height:30 align:#center offset:[0,0] toolTip:"Belirlediginiz Parametreler Dahilinde Sisteminizi Olusturur."
on Olustur pressed do
(
Global SisGen=SistemGenis.value--Sistem genisligi (Kolon merkezlerinden)
Global SisUz=SistemUzun.value--Sistem uzunlugu (Kolon merkezlerinden)
Global KolDist=SisUz/(KolonSayisi.value-1)--Kolonlar arasi mesafe (merkezlerinden)
Global KolSay=KolonSayisi.value--Sistemin tek siradaki kolon sayisi
Global KolEni=SisGen/50--Kolon eni
Global KolKal=KolDist/40--Kolon kalinligi
Global KolYuks=KolonYuks.value--Kolon yüksekligi
Global KirYuks=SisGen/16--Kiris yüksekligi
Global KirGen=SisGen-KolEni--Kiris genisligi
Global KirAltH=KolYuks-KirYuks--Kirisin yerden yüksekligi
Global KirHoPrYuk=SisGen/200--Kiris yatay profil yükseklikleri(kalinliklari)
```

```
Global KirVePrEni=KirHoPrYuk/2--Kiris düsey profil enleri(kalinliklari)
Global KirVePrYuk=KirYuks-(2*KirHoPrYuk)--Kiris düsey profil yükseklikleri
Global VerPrSayKS=(Pow (Pow SisGen 2) 0.2)/(Pow (Pow 1000 2) 0.2)--Kiris düsey profil sayisi KatSayisi (Sistem genisligi disinda, profillerin sikligini arttirip azaltmak için kullanilan bir parametre..)
Global VerPrSay=((((KirGen/2)-(KirHoPrYuk+(KirVePrEni/2)))/KirVePrYuk)*VerPrSayKS)-1--Kiris düsey profil sayisi nin yarisi
Global VerPrSay = (VerPrSay as integer)--Sayiyi tam sayi yaptik...
Global VerPrDist=((KirGen/2)-(KirHoPrYuk+(KirVePrEni/2))-(VerPrSay*KirVePrEni))/(VerPrSay+1)--Kiris düsey profiller arasi mesafe
Global VerPrDist2=((KirGen/2)-(KirHoPrYuk+(KirVePrEni/2)))/(VerPrSay+1)--Kiris düsey profiller arasi mesafe (Warren Tipleri için
Global DiagPrUz=sqrt ((Pow KirVePrYuk 2) + (Pow VerPrDist 2))--Kiris diagonal profil uzunlugu
Global DiagPrXPo=(DiagPrUz*(KirVePrEni/2))/KirVePrYuk--Kiris diagonal profil köseden X yönündeki mesafesi
Global DiagPrYpo=KirVePrYuk*DiagPrXPo/VerPrDist--Kiris diagonal profil köseden Y yönündeki mesafesi
Global theModi = extrude amount:(KolKal/4)
if (KolDist) > 1000 or (KolDist) < 300 then messagebox "DIKKAT!!! Girilen parametler uygun degil. Parametrelerinizi gözden geçiriniz... (Kolonlar arasi mesafe 300cm ile 1000cm arasinda olmali)"
else
(
if (KolYuks-KirYuks) < 300 then messagebox "DIKKAT!!! Girilen parametler uygun degildir. Parametrelerinizi gözden geçiriniz... (Kiris yerden yüksekligi minumum 300cm olmali)"
else
((
CamDis = Targetcamera fov:40 nearclip:1 farclip:1000 nearrange:0 farrange:1000 mpassEnabled:off mpassRenderPerPass:off name:"CameraDis"
$CameraDis.target = Targetobject transform:(matrix3 [1,0,0] [0,1,0] [0,0,1] [SisGen/2, SisUz/10, KolYuks/2])
$CameraDis.pos = [4*SisGen/3, -(2*((13*SisGen/5))/3), 200]
CamIc = Targetcamera fov:65 nearclip:1 farclip:1000 nearrange:0 farrange:1000 mpassEnabled:off mpassRenderPerPass:off name:"CameraIc"
$CameraIc.target = Targetobject transform:(matrix3 [1,0,0] [0,1,0] [0,0,1] [SisGen/2, 4*SisUz/5, KolYuks/2])
$CameraIc.pos = [5*SisGen/6, -500, 200]
)
case RenderSistem.radbut_RenderType.state of  --Render ve malzeme tipini seçer
(
1: (renderpresets.LoadAll 0 (SetFilePath + "Standard.rps")
--Steel1
meditMaterials[1] = Standardmaterial ()
meditMaterials[1].name = "Steel1"
meditMaterials[1].adLock = off
meditMaterials[1].ambient = color 80 80 80
meditMaterials[1].Diffuse = color 110 120 130
meditMaterials[1].specularLevel = 70
meditMaterials[1].glossiness = 50

--Steel2
meditMaterials[2] = Standardmaterial ()
meditMaterials[2].name = "Steel2"
meditMaterials[2].adLock = off
meditMaterials[2].ambient = color 80 80 80
meditMaterials[2].Diffuse = color 165 170 180
meditMaterials[2].specularLevel = 70
meditMaterials[2].glossiness = 50
--Roof
meditMaterials[7] = Standardmaterial ()
meditMaterials[7].name = "Roof"
meditMaterials[7].adLock = off
meditMaterials[7].ambient = color 80 80 80
meditMaterials[7].Diffuse = color 100 0 0
meditMaterials[7].specularLevel = 55
meditMaterials[7].glossiness = 30
--Panels
meditMaterials[8] = Standardmaterial ()
meditMaterials[8].name = "Panel"
meditMaterials[8].adLock = off
meditMaterials[8].ambient = color 80 80 80
meditMaterials[8].Diffuse = color 200 200 200
meditMaterials[8].specularLevel = 55
meditMaterials[8].glossiness = 30
--Ground
meditMaterials[13] = MatteShadow ()
meditMaterials[13].name = "Ground"
meditMaterials[13].opaqueAlpha = on
meditMaterials[13].applyAtmosphere = on
--BackGround için Vray
meditMaterials[14] = Standardmaterial ()
meditMaterials[14].name = "BackGroundVR"
meditMaterials[14].diffuseMap = Gradient ()
meditMaterials[14].diffuseMap.coordinates.mappingType = 1
meditMaterials[14].diffuseMap.coordinates.mapping = 3
meditMaterials[14].diffuseMap.name = "BG_Gradient"
meditMaterials[14].diffuseMap.color1 = color 0 72 255
meditMaterials[14].diffuseMap.color2 = color 201 223 223
meditMaterials[14].diffuseMap.color3 = color 255 254 237
meditMaterials[14].diffuseMap.map1 = Noise ()
meditMaterials[14].diffuseMap.map1.name = "BG-GR_Noise"
meditMaterials[14].diffuseMap.map1.color1 = color 0 62 154
meditMaterials[14].diffuseMap.map1.color2 = color 255 244 201
meditMaterials[14].diffuseMap.map1.size = 18
meditMaterials[14].diffuseMap.map1.phase = 7
--Environment Map ( BackGroundun diffuse unu kullaniyor)
environmentMap = Gradient ()
environmentMap = meditMaterials[14].diffuseMap
)
2: (renderpresets.LoadAll 0 (SetFilePath + "Vray.rps")
--Steel1
meditMaterials[1] = VRayMtl ()
meditMaterials[1].name = "Steel1"
meditMaterials[1].Diffuse = color 110 120 130
meditMaterials[1].Reflection = color 20 20 20
meditMaterials[1].reflection_lockGlossiness = off
meditMaterials[1].hilight_glossiness = 0.9
meditMaterials[1].reflection_glossiness = 0.9
--Steel2
meditMaterials[2] = VRayMtl ()
meditMaterials[2].name = "Steel2"
meditMaterials[2].Diffuse = color 165 170 180
meditMaterials[2].Reflection = color 20 20 20
meditMaterials[2].reflection_lockGlossiness = off
meditMaterials[2].hilight_glossiness = 0.9
meditMaterials[2].reflection_glossiness = 0.9
--Roof
meditMaterials[7] = VRayMtl ()
meditMaterials[7].name = "Roof"
meditMaterials[7].Diffuse = color 100 0 0
meditMaterials[7].Reflection = color 20 20 20
meditMaterials[7].reflection_lockGlossiness = off
meditMaterials[7].hilight_glossiness = 0.65
meditMaterials[7].reflection_glossiness = 0.9
--Panels
meditMaterials[8] = VRayMtl ()
meditMaterials[8].name = "Panel"
meditMaterials[8].Diffuse = color 200 200 200
meditMaterials[8].Reflection = color 20 20 20
meditMaterials[8].reflection_lockGlossiness = off
meditMaterials[8].hilight_glossiness = 0.65
meditMaterials[8].reflection_glossiness = 0.9
--Ground
meditMaterials[13] = VRayMtlWrapper ()
meditMaterials[13].name = "Ground"
meditMaterials[13].matteSurface = on
meditMaterials[13].matte_shadows = on
meditMaterials[13].matte_shadowsAffectAlpha = on
meditMaterials[13].baseMtl = VRayMtl ()
meditMaterials[13].baseMtl.name = "GroundVR"
meditMaterials[13].baseMtl.Diffuse = color 255 255 255
--BackGround için Vray
meditMaterials[14] = VRayMtl ()
meditMaterials[14].texmap_diffuse = Gradient ()
meditMaterials[14].texmap_diffuse.coordinates.mappingType = 1
meditMaterials[14].texmap_diffuse.coordinates.mapping = 3
meditMaterials[14].texmap_diffuse.color1 = color 0 72 255
meditMaterials[14].texmap_diffuse.color2 = color 201 223 223
meditMaterials[14].texmap_diffuse.color3 = color 255 254 237
meditMaterials[14].texmap_diffuse.noiseSize = 0
meditMaterials[14].texmap_diffuse.map1 = Noise ()
meditMaterials[14].texmap_diffuse.map1.color1 = color 0 62 154
meditMaterials[14].texmap_diffuse.map1.color2 = color 255 244 201
meditMaterials[14].texmap_diffuse.map1.size = 18
meditMaterials[14].texmap_diffuse.map1.phase = 7
meditMaterials[14].name = "BackGroundVR"
meditMaterials[14].texmap_diffuse.name = "BG_Gradient"
meditMaterials[14].texmap_diffuse.map1.name = "BG-GR_Noise"
--Environment Map ( BackGroundun diffuse unu kullaniyor)
environmentMap = Gradient ()
environmentMap = meditMaterials[14].texmap_diffuse
))
fn KirSist =
(
--Kolonlarin olusturulmasi
KolonSol = box width:KolEni length:KolKal height:KolYuks name: "KolonSol" wirecolor:gray
KolonSag = instance KolonSol name:"KolonSag" wirecolor:gray
Move KolonSag [SisGen, 0, 0]
--Kiris Olusturma
KirAlt = box width:KirGen length:(KolKal/2) height:KirHoPrYuk name: "KirisAltProfil" wirecolor:orange --Alt baslik
Move KirAlt [(SisGen/2), 0, KirAltH]
KirUst = instance KirAlt name: "KirisUstProfil" wirecolor:orange --Üst baslik
move KirUst [0, 0, (KirYuks-KirHoPrYuk)]
KirPrSol = box width:KirHoPrYuk length:(KolKal/2) height:KirVePrYuk name: "KirisSolProfil" wirecolor:orange --Baslik sol profil
Move KirPrSol [(KolEni/2)+(KirHoPrYuk/2), 0, KirAltH+KirHoPrYuk]
KirPrSag = instance KirPrSol name: "KirisSagProfil"  wirecolor:orange --Baslik sag profil
Move KirPrSag [KirGen-KirHoPrYuk, 0, 0]
for i = 0 to (2*VerPrSay) do
(
```

```
KirPrVer = box width:KirVePrEni length:(KolKal/4) height:KirVePrYuk name: "KirisVerticalProfil" wirecolor:white --Vertical profiller
Move KirPrVer [((KolEni/2)+KirHoPrYuk+(KirVePrEni/2)+VerprDist), 0, KirAltH+KirHoPrYuk]
Move KirPrVer [(VerprDist+KirVePrEni)*i,0,0]
)
for m = 0 to VerPrSay do
(
DiagPrA = SplineShape pos:[0,0,0] name: "KirisDiagonalProfilA" --Diagonal Profiller
addNewSpline DiagPrA
addKnot DiagPrA 1 #corner #line [0,0,0]
addKnot DiagPrA 1 #corner #line [DiagPrXPo,0,0]
addKnot DiagPrA 1 #corner #line [VerPrDist,(KirVePrYuk-DiagPrYPo),0]
addKnot DiagPrA 1 #corner #line [VerPrDist,KirVePrYuk,0]
addKnot DiagPrA 1 #corner #line [(VerPrDist-DiagPrXPo),KirVePrYuk,0]
addKnot DiagPrA 1 #corner #line [0,DiagPrYPo,0]
Close DiagPrA 1
DiagPrA.wirecolor = color 255 255 255
rotate DiagPrA (eulerangles 90 0 0)
updateShape DiagPrA
Move DiagPrA [(KolEni/2)+KirHoPrYuk, (KolKal/8), KirAltH+KirHoPrYuk]
Move DiagPrA [(2*(VerprDist+KirVePrEni))*m,0,0]
addmodifier DiagPrA theModi
)for n = 0 to VerPrSay do
(
DiagPrB = SplineShape pos:[0,0,0] name: "KirisDiagonalProfilB" --Diagonal Profiller
addNewSpline DiagPrB
addKnot DiagPrB 1 #corner #line [0,0,0]
addKnot DiagPrB 1 #corner #line [-DiagPrXPo,0,0]
addKnot DiagPrB 1 #corner #line [-VerPrDist,(KirVePrYuk-DiagPrYPo),0]
addKnot DiagPrB 1 #corner #line [-VerPrDist,KirVePrYuk,0]
addKnot DiagPrB 1 #corner #line [-(VerPrDist-DiagPrXPo),KirVePrYuk,0]
addKnot DiagPrB 1 #corner #line [0,DiagPrYPo,0]
Close DiagPrB 1
DiagPrB.wirecolor = color 255 255 255
rotate DiagPrB (eulerangles 90 0 0)
updateShape DiagPrB
Move DiagPrB [(KolEni/2)+KirHoPrYuk+(2*VerPrDist)+KirVePrEni, (KolKal/8), KirAltH+KirHoPrYuk]
Move DiagPrB [(2*(VerprDist+KirVePrEni))*n,0,0]
addmodifier DiagPrB theModi
)
) --Fn KirSist Kapama
for t = 0 to KolSay-1 do--Kiris-Kolon sistemimizi burada tamamliyoruz ve tüm akslara kopyaliyoruz (instance olarak)
(
KirSist()
$kirisVerticalProfil*.baseobject = $kirisVerticalProfil*[1].baseobject
$KirisDiagonalProfilA*.baseobject = $KirisDiagonalProfilA*[1].baseobject
$KirisDiagonalProfilB*.baseobject = $KirisDiagonalProfilB*[1].baseobject
$KolonSol*.baseobject = $KolonSol*[1].baseobject
$KolonSag*.baseobject = $KolonSag*[1].baseobject
$KirisAltProfil*.baseobject = $KirisAltProfil*[1].baseobject
$KirisUstProfil*.baseobject = $KirisUstProfil*[1].baseobject
$KirisSolProfil*.baseobject = $KirisSolProfil*[1].baseobject
$KirisSagProfil*.baseobject = $KirisSagProfil*[1].baseobject
$KolonS*.material = meditMaterials[1]
$Kiris*.material = meditMaterials[2]
KirisSistem = group geometry name:"KirisSistem" select:false
Move KirisSistem [0, KolDist*t, 0]
)
for v = 0 to KolSay-2 do--Kiris sistemleri arasindaki düz profil krisleri yerlestirdik (yine instance olarak)
(
KirDuzSol = box width:(KolEni/2) length:(KolDist-KolKal) height:(KolDist/16) pos:[(-KolEni/4), (KolDist/2), (KolYuks-(KolDist/16))] name: "KirisDuzSol" Wirecolor:white
KirDuzSag = instance KirDuzSol name: "KirisDuzSag" Wirecolor:white
Move KirDuzSag [(Sisgen+(KolEni/2)), 0, 0]
Move KirDuzSol [0,KolDist*v,0]
Move KirDuzSag [0,KolDist*v,0]
$KirisDuzSol*.baseobject = $KirisDuzSol*[1].baseobject
$KirisDuzSag*.baseobject = $KirisDuzSag*[1].baseobject


$KirisDuz*.material = meditMaterials[2]
)
if SistemGorunus.chbox_duvar.checked then
(
for y = 0 to (2*KolSay)-5 do--Yan paneller - Ortadakiler (instance olarak)
(
PanelSol = box width:10 length:((KolDist/2)-2) height:KolYuks pos:[(-KolEni/2)-5, (KolDist/2)+(KolDist/4), 0] name: "PanelSol" Wirecolor:(color 100 150 200)
PanelSag = instance PanelSol name: "PanelSag" Wirecolor:(color 100 150 200)
Move PanelSag [(Sisgen+KolEni+10), 0, 0]
Move PanelSol [0,(KolDist/2)*y,0]
Move PanelSag [0,(KolDist/2)*y,0]
$PanelSol*.baseobject = $PanelSol*[1].baseobject
$PanelSag*.baseobject = $PanelSag*[1].baseobject
$Panel*.material = meditMaterials[8]
)
for z = 0 to 1 do--Yan paneller - Bastakiler (instance olarak)
(
PanelSol1 = box width:10 length:((KolDist/2)-2+(KolKal/2)+1) height:KolYuks pos:[(-KolEni/2)-5, (KolDist/4)-(KolKal/4)-0.5, 0] name: "PanelSol1" Wirecolor:(color 100 150 200)
PanelSag1 = instance PanelSol1 name: "PanelSag1" Wirecolor:(color 100 150 200)
Move PanelSag1 [(Sisgen+KolEni+10), 0, 0]
Move PanelSol1 [0,(SisUz-(KolDist/2)+8.5)*z,0]
Move PanelSag1 [0,(SisUz-(KolDist/2)+8.5)*z,0]
$PanelSol1*.baseobject = $PanelSol1*[1].baseobject
$PanelSag1*.baseobject = $PanelSag1*[1].baseobject
$Panel*.material = meditMaterials[8]
))
Zemin = plane width:(SisGen*3) length:(SisUz*3) pos:[(SisGen/2), (SisUz/2), 0] name:"Zemin" wirecolor:White --Zemin olusturur
$Zemin*.material = meditMaterials[13]
if SistemGorunus.chbox_cati.checked then --Çatiyi olusturur duruma göre
Cati = box width:(SisGen+KolEni+20) length:(SisUz+KolKal) height:20 pos:[(SisGen/2), (SisUz/2), KolYuks] name: "Cati" wirecolor:(color 150 0 0)
$Cati*.material = meditMaterials[7]
case RenderSistem.radbut_Cam_type.state of --Kameralari olusturur ve seçimini yapar
(
1: (select $CameraDis)
2: (select $CameraIc)
)
max vpt camera
deselect objects
backgroundColor = color 255 255 255
if RenderSistem.chbox_render.checked and (RenderSistem.radbut_render_time.state == 2) then --Render seçimini yapar ve render alir
max quick render
if (RenderSistem.radbut_render_time.state == 1) then --Render alir ama önce sorar
if queryBox "Sonra hatirlat demistin. Render alayim mi?" beep:false then
max quick render
) --If-Else kapama
) --If-Else kapama

) --On-do kapama
on buttGeri pressed do
(
addRollout bilgiParalel PSS
addRollout ParalelKafesKiris PSS
removeRollout bilgiParalelWarren PSS
removeRollout SisParaPlWarren PSS
removeRollout SistemGorunus PSS
removeRollout RenderSistem PSS
removeRollout Exporting PSS
)
) --Rollout kapama
```

ParalelWARREN_Modified.ms

```
--PSS = newRolloutFloater "P.S.S. 1.01 Beta" 240 770 800 100
GetFilePath = getThisScriptFilename()
SetFilePath= getFileNamePath GetFilePath
rollout bilgiParalelWarrenM "Kisa Bilgi" RolledUp:true
(
group "Script Kullanimi"
(
label lab1 "--Asagidaki paralel baslikli WARREN-Modified" align:#left
label lab2 "      kafes kirisi sistem özelliklerinizi seçiniz" align:#left
label lab3 "      ve render - malzeme ayarlarini yapiniz." align:#left
))
rollout SisParaPlWarrenM "Sistem Parametreleri"
(
button buttGeri "<<< Üst Menüye Geri Dön <<<" border:true width:220 height:30 align:#center toolTip:"Bir Önceki Menüye Geri Döner"
group "Açikliklar"
(
spinner SistemGenis "Sistem Genisligi (cm):" range:[500,5000,2000] type:#float toolTip:"Sistemin Geçecegi Mesafeyi Belirler. (5m-50m)"
spinner SistemUzun "Sistem Uzunlugu (cm):" range:[500,30000,3000] type:#float toolTip:"Yapinin Toplam Uzunlugunu Belirler. (5m-300m)"
)
group "Kolonlar"
(
spinner KolonYuks "Kolon Yüksekligi (cm):" range:[400,2000,800] type:#float toolTip:"Sistemin Yerden En Yüksek Noktasina Olan Mesafeyi Belirler. (4m-20m)"
spinner KolonSayisi "Kolon Sayisi:" range:[2,33,6] type:#integer toolTip:"Sistem Uzunlugu Yönündeki Toplam Aks Sayisini Belirler. (2-33)"
)
group "Diger Anlik Parametrik Bilgiler"
(
label labX1 "Kiris Yüksekligi: 125.0 cm" align:#left
label labX2 "Kirisin Yerden Yüksekligi: 675.0 cm" align:#left
label labX3 "Kolonlar Arasi Mesafe: 600.0 cm" align:#left
)
fn Paraminfo =
(
labX1.text = "Kiris Yüksekligi: " + (SistemGenis.value/16) as string + " cm"
labX2.text = "Kirisin Yerden Yüksekligi: " + (KolonYuks.value-(SistemGenis.value/16)) as string + " cm"
```

```
labX3.text = "Kolonlar Arasi Mesafe: " + ((SistemUzun.value)/(KolonSayisi.value-1)) as string + " cm"
)
on SistemGenis changed value do  Paraminfo()
on SistemUzun changed value do Paraminfo()
on KolonYuks changed value do Paraminfo()
on KolonSayisi changed value do Paraminfo()
button sil "Sil" border:true width:135 height:20 align:#left offset:[-10,0] across:2 toolTip:"Sahnedeki Bütün Nesneleri Siler."
on sil pressed do delete objects
button resett "Reset" border:true width:80 height:20 align:#right offset:[10,0] toolTip:"Sahneye Reset Atar (Yeniden Baslatir)."
on resett pressed do resetMaxFile()
button Olustur "Olustur" border:true width:220 height:30 align:#center offset:[0,0] toolTip:"Belirlediginiz Parametreler Dahilinde Sisteminizi Olusturur."
on Olustur pressed do
(Global SisGen=SistemGenis.value--Sistem genisligi (Kolon merkezlerinden)
Global SisUz=SistemUzun.value--Sistem uzunlugu (Kolon merkezlerinden)
Global KolDist=SisUz/(KolonSayisi.value-1)--Kolonlar arasi mesafe (merkezlerinden)
Global KolSay=KolonSayisi.value--Sistemin tek siradaki kolon sayisi
Global KolEni=SisGen/50--Kolon eni
Global KolKal=KolDist/40--Kolon kalinligi
Global KolYuks=KolonYuks.value--Kolon yüksekligi
Global KirYuks=SisGen/16--Kiris yüksekligi
Global KirGen=SisGen+KolEni--Kiris genisligi
Global KirAltH=KolYuks-KirYuks--Kirisin yerden yüksekligi
Global KirHoPrYuk=SisGen/200--Kiris yatay profil yüksekliklerı(kalinliklari)
Global KirVePrEni=KirHoPrYuk/2--Kiris düsey profil enleri(kalinliklari)
Global KirVePrYuk=KirYuks-(2*KirHoPrYuk)--Kiris düsey profil yüksekliklerı
Global VerPrSayKS=(Pow (Pow SisGen 2) 0.2)/(Pow (Pow 1000 2) 0.2)--Kiris düsey profil sayisi KatSayisi (Sistem genisligi disinda, profillerin sikligini arttirip azaltmak için kullanilan bir parametre..)
Global VerPrSay=((((KirGen/2)-(KirVePrEni/2))/KirVePrYuk)*VerPrSayKS)-1--Kiris düsey profil sayisi nin yarisi
Global VerPrSay = (VerPrSay as integer)--Sayiyi tam sayi yaptik...
Global VerPrDist=((KirGen/2)-(KirVePrEni/2)-(VerPrSay*KirVePrEni))/(VerPrSay+1)--Kiris düsey profiller arasi mesafe
Global DiagPrUz=sqrt ((Pow KirVePrYuk 2) + (Pow VerPrDist 2))--Kiris diagonal profil uzunlugu
Global DiagPrXPo=(DiagPrUz*(KirVePrEni/2))/KirVePrYuk--Kiris diagonal profil köseden X yönündeki mesafesi
Global DiagPrYpo=KirVePrYuk*DiagPrXPo/VerPrDist--Kiris diagonal profil köseden Y yönündeki mesafesi
Global theModi = extrude amount:(KolKal/4)
Global theModi2 = extrude amount:(KolKal/2)
Global DiagPr1XPo=VerPrDist-((VerPrDist*KirVePrYuk)/(KirHoPrYuk+KirVePrYuk))--Kiris diagonal profil köseden X yönündeki mesafesi
if (KolDist) > 1000 or (KolDist) < 300 then messagebox "DIKKAT!!! Girilen parametler uygun degil. Parametrelerinizi gözden geçiriniz... (Kolonlar arasi mesafe 300cm ile 1000cm arasinda olmali)"
else
(
if (KolYuks-KirYuks) < 300 then messagebox "DIKKAT!!! Girilen parametler uygun degildir. Parametrelerinizi gözden geçiriniz... (Kiris yerden yüksekligi minumum 300cm olmali)"
else
((
CamDis = Targetcamera fov:40 nearclip:1 farclip:1000 nearrange:0 farrange:1000 mpassEnabled:off mpassRenderPerPass:off name:"CameraDis"
$CameraDis.target = Targetobject transform:(matrix3 [1,0,0] [0,1,0] [0,0,1] [SisGen/2, SisUz/10, KolYuks/2])
$CameraDis.pos = [4*SisGen/3, -(2*((13*SisGen/5))/3), 200]
CamIc = Targetcamera fov:65 nearclip:1 farclip:1000 nearrange:0 farrange:1000 mpassEnabled:off mpassRenderPerPass:off name:"CameraIc"
$CameraIc.target = Targetobject transform:(matrix3 [1,0,0] [0,1,0] [0,0,1] [SisGen/2, 4*SisUz/5, KolYuks/2])
$CameraIc.pos = [5*SisGen/6, -500, 200]
)
case RenderSistem.radbut_RenderType.state of  --Render ve malzeme tipini seçer
(
1: (renderpresets.LoadAll 0 (SetFilePath + "Standard.rps")
--Steel1
meditMaterials[1] = Standardmaterial ()
meditMaterials[1].name = "Steel1"
meditMaterials[1].adLock = off
meditMaterials[1].ambient = color 80 80 80
meditMaterials[1].Diffuse = color 110 120 130
meditMaterials[1].specularLevel = 70
meditMaterials[1].glossiness = 50
--Steel2
meditMaterials[2] = Standardmaterial ()
meditMaterials[2].name = "Steel2"
meditMaterials[2].adLock = off
meditMaterials[2].ambient = color 80 80 80
meditMaterials[2].Diffuse = color 165 170 180
meditMaterials[2].specularLevel = 70
meditMaterials[2].glossiness = 50
--Roof
meditMaterials[7] = Standardmaterial ()
meditMaterials[7].name = "Roof"
meditMaterials[7].adLock = off
meditMaterials[7].ambient = color 80 80 80
meditMaterials[7].Diffuse = color 100 0 0
meditMaterials[7].specularLevel = 55
meditMaterials[7].glossiness = 30
--Panels
meditMaterials[8] = Standardmaterial ()
meditMaterials[8].name = "Panel"
meditMaterials[8].adLock = off
meditMaterials[8].ambient = color 80 80 80
meditMaterials[8].Diffuse = color 200 200 200
meditMaterials[8].specularLevel = 55
meditMaterials[8].glossiness = 30
--Ground
meditMaterials[13] = MatteShadow ()
meditMaterials[13].name = "Ground"
meditMaterials[13].opaqueAlpha = on
meditMaterials[13].applyAtmosphere = on
--BackGround için Vray
meditMaterials[14] = Standardmaterial ()
meditMaterials[14].name = "BackGroundVR"
meditMaterials[14].diffuseMap = Gradient ()
meditMaterials[14].diffuseMap.coordinates.mappingType = 1
meditMaterials[14].diffuseMap.coordinates.mapping = 3
meditMaterials[14].diffuseMap.name = "BG_Gradient"
meditMaterials[14].diffuseMap.color1 = color 0 72 255
meditMaterials[14].diffuseMap.color2 = color 201 223 223
meditMaterials[14].diffuseMap.color3 = color 255 254 237
meditMaterials[14].diffuseMap.map1 = Noise ()
meditMaterials[14].diffuseMap.map1.name = "BG-GR_Noise"
meditMaterials[14].diffuseMap.map1.color1 = color 0 62 154
meditMaterials[14].diffuseMap.map1.color2 = color 255 244 201
meditMaterials[14].diffuseMap.map1.size = 18
meditMaterials[14].diffuseMap.map1.phase = 7
--Environment Map ( BackGroundun diffuse unu kullaniyor)
environmentMap = Gradient ()
environmentMap = meditMaterials[14].diffuseMap
)
2: (renderpresets.LoadAll 0 (SetFilePath + "Vray.rps")
--Steel1
meditMaterials[1] = VRayMtl ()
meditMaterials[1].name = "Steel1"
meditMaterials[1].Diffuse = color 110 120 130
meditMaterials[1].Reflection = color 20 20 20
meditMaterials[1].reflection_lockGlossiness = off
meditMaterials[1].hilight_glossiness = 0.9
meditMaterials[1].reflection_glossiness = 0.9
--Steel2
meditMaterials[2] = VRayMtl ()
meditMaterials[2].name = "Steel2"
meditMaterials[2].Diffuse = color 165 170 180
meditMaterials[2].Reflection = color 20 20 20
meditMaterials[2].reflection_lockGlossiness = off
meditMaterials[2].hilight_glossiness = 0.9
meditMaterials[2].reflection_glossiness = 0.9
--Roof
meditMaterials[7] = VRayMtl ()
meditMaterials[7].name = "Roof"
meditMaterials[7].Diffuse = color 100 0 0
meditMaterials[7].Reflection = color 20 20 20
meditMaterials[7].reflection_lockGlossiness = off
meditMaterials[7].hilight_glossiness = 0.65
meditMaterials[7].reflection_glossiness = 0.9
--Panels
meditMaterials[8] = VRayMtl ()
meditMaterials[8].name = "Panel"
meditMaterials[8].Diffuse = color 200 200 200
meditMaterials[8].Reflection = color 20 20 20
meditMaterials[8].reflection_lockGlossiness = off
meditMaterials[8].hilight_glossiness = 0.65
meditMaterials[8].reflection_glossiness = 0.9

--Ground
meditMaterials[13] = VRayMtlWrapper ()
meditMaterials[13].name = "Ground"
meditMaterials[13].matteSurface = on
meditMaterials[13].matte_shadows = on
meditMaterials[13].matte_shadowsAffectAlpha = on
meditMaterials[13].baseMtl = VRayMtl ()
meditMaterials[13].baseMtl.name = "GroundVR"
meditMaterials[13].baseMtl.Diffuse = color 255 255 255
--BackGround için Vray
meditMaterials[14] = VRayMtl ()
meditMaterials[14].texmap_diffuse = Gradient ()
meditMaterials[14].texmap_diffuse.coordinates.mappingType = 1
meditMaterials[14].texmap_diffuse.coordinates.mapping = 3
meditMaterials[14].texmap_diffuse.color1 = color 0 72 255
meditMaterials[14].texmap_diffuse.color2 = color 201 223 223
meditMaterials[14].texmap_diffuse.color3 = color 255 254 237
meditMaterials[14].texmap_diffuse.noiseSize = 0
meditMaterials[14].texmap_diffuse.map1 = Noise ()
meditMaterials[14].texmap_diffuse.map1.color1 = color 0 62 154
meditMaterials[14].texmap_diffuse.map1.color2 = color 255 244 201
meditMaterials[14].texmap_diffuse.map1.size = 18
meditMaterials[14].texmap_diffuse.map1.phase = 7
```

```
meditMaterials[14].name = "BackGroundVR"
meditMaterials[14].texmap_diffuse.name = "BG_Gradient"
meditMaterials[14].texmap_diffuse.map1.name = "BG-GR_Noise"
--Environment Map ( BackGroundun diffuse unu kullaniyor)
environmentMap = Gradient ()
environmentMap = meditMaterials[14].texmap_diffuse
))
fn KirSist =
(
--Kolonlarin olusturulmasi
KolonSol = box width:KolEni length:KolKal height:KirAltH name: "KolonSol" wirecolor:gray
KolonSag = instance KolonSol name:"KolonSag" wirecolor:gray
Move KolonSag [SisGen, 0, 0]
--Kiris Olusturma
KirAlt = box width:KirGen length:(KolKal/2) height:KirHoPrYuk name: "KirisAltProfil" wirecolor:orange --Alt baslik
Move KirAlt [(SisGen/2), 0, KirAltH]
KirUst = copy KirAlt name: "KirisUstProfil" wirecolor:orange --Üst baslik
KirUst.width = (KirGen-(2*VerPrDist))
move KirUst [0, 0, (KirYuks-KirHoPrYuk)]
for i = 0 to (2*VerPrSay) do
(
KirPrVer = box width:KirVePrEni length:(KolKal/4) height:KirVePrYuk name: "KirisVerticalProfil" wirecolor:white --Vertical profiller
Move KirPrVer [((KirVePrEni/2)+VerprDist-(KolEni/2)), 0, KirAltH+KirHoPrYuk]
Move KirPrVer [(VerprDist+KirVePrEni)*i,0,0]
)
DiagPrSol = SplineShape pos:[0,0,0] name: "KirisDiagonalProfilSolBas" --Diagonal Profil Sol Bastaki
addNewSpline DiagPrSol
addKnot DiagPrSol 1 #corner #line [0,0,0]
addKnot DiagPrSol 1 #corner #line [DiagPr1XPo,0,0]
addKnot DiagPrSol 1 #corner #line [VerPrDist,KirVePrYuk,0]
addKnot DiagPrSol 1 #corner #line [VerPrDist,KirVePrYuk+KirHoPrYuk,0]
Close DiagPrSol 1
DiagPrSol.wirecolor = orange
rotate DiagPrSol (eulerangles 90 0 0)
updateShape DiagPrSol
Move DiagPrSol [-(KolEni/2), (KolKal/4), KirAltH+KirHoPrYuk]
addmodifier DiagPrSol theModi2
DiagPrSag = SplineShape pos:[0,0,0] name: "KirisDiagonalProfilSagBas" --Diagonal Profil Sag Bastaki
addNewSpline DiagPrSag
addKnot DiagPrSag 1 #corner #line [0,0,0]
addKnot DiagPrSag 1 #corner #line [-DiagPr1XPo,0,0]
addKnot DiagPrSag 1 #corner #line [-VerPrDist,KirVePrYuk,0]
addKnot DiagPrSag 1 #corner #line [-VerPrDist,KirVePrYuk+KirHoPrYuk,0]
Close DiagPrSag 1
DiagPrSag.wirecolor = orange
rotate DiagPrSag (eulerangles 90 0 0)
updateShape DiagPrSag
Move DiagPrSag [KirGen-(KolEni/2), (KolKal/4), KirAltH+KirHoPrYuk]
addmodifier DiagPrSag theModi2
for m = 1 to VerPrSay do
(
DiagPrA = SplineShape pos:[0,0,0] name: "KirisDiagonalProfilA" --Diagonal Profiller
addNewSpline DiagPrA
addKnot DiagPrA 1 #corner #line [0,0,0]
addKnot DiagPrA 1 #corner #line [DiagPrXPo,0,0]
addKnot DiagPrA 1 #corner #line [VerPrDist,(KirVePrYuk-DiagPrYPo),0]
addKnot DiagPrA 1 #corner #line [VerPrDist,KirVePrYuk,0]
addKnot DiagPrA 1 #corner #line [(VerPrDist-DiagPrXPo),KirVePrYuk,0]
addKnot DiagPrA 1 #corner #line [0,DiagPrYPo,0]
Close DiagPrA 1
DiagPrA.wirecolor = color 255 255 255
rotate DiagPrA (eulerangles 90 0 0)
updateShape DiagPrA
Move DiagPrA [-(KolEni/2), (KolKal/8), KirAltH+KirHoPrYuk]
Move DiagPrA [(2*(VerPrDist+KirVePrEni))*m,0,0]
addmodifier DiagPrA theModi
)
for n = 1 to VerPrSay do
(
DiagPrB = SplineShape pos:[0,0,0] name: "KirisDiagonalProfilB" --Diagonal Profiller
addNewSpline DiagPrB
addKnot DiagPrB 1 #corner #line [0,0,0]
addKnot DiagPrB 1 #corner #line [-DiagPrXPo,0,0]
addKnot DiagPrB 1 #corner #line [-VerPrDist,(KirVePrYuk-DiagPrYPo),0]
addKnot DiagPrB 1 #corner #line [-VerPrDist,KirVePrYuk,0]
addKnot DiagPrB 1 #corner #line [-(VerPrDist-DiagPrXPo),KirVePrYuk,0]
addKnot DiagPrB 1 #corner #line [0,DiagPrYPo,0]
Close DiagPrB 1
DiagPrB.wirecolor = color 255 255 255
rotate DiagPrB (eulerangles 90 0 0)
updateShape DiagPrB
Move DiagPrB [-(KolEni/2)-KirVePrEni, (KolKal/8), KirAltH+KirHoPrYuk]
Move DiagPrB [(2*(VerPrDist+KirVePrEni))*n,0,0]
addmodifier DiagPrB theModi
)) --Fn KirSist Kapama
for t = 0 to KolSay-1 do--Kiris-Kolon sistemimizi burada tamamliyoruz ve tüm akslara kopyaliyoruz (instance olarak)
(
KirSist()
$kirisVerticalProfil*.baseobject = $kirisVerticalProfil*[1].baseobject
$KirisDiagonalProfilA*.baseobject = $KirisDiagonalProfilA*[1].baseobject
$KirisDiagonalProfilB*.baseobject = $KirisDiagonalProfilB*[1].baseobject
$KolonSol*.baseobject = $KolonSol*[1].baseobject
$KolonSag*.baseobject = $KolonSag*[1].baseobject
$KolonS*.material = meditMaterials[1]
$Kiris*.material = meditMaterials[2]
KirisSistem = group geometry name:"KirisSistem" select:false
Move KirisSistem [0, KolDist*t, 0]
)
for v = 0 to KolSay-2 do--Kiris sistemleri arasindaki düz profil krisleri yerlestirdik (yine instance olarak)
(
KirDuzSol = box width:(KolEni/2) length:(KolDist-KolKal) height:(KolDist/16) pos:[(-KolEni/4), (KolDist/2), (KirAltH-(KolDist/16))] name: "KirisDuzSol" Wirecolor:white
KirDuzSag = instance KirDuzSol name: "KirisDuzSag" Wirecolor:white
Move KirDuzSag [(Sisgen+(KolEni/2)), 0, 0]
Move KirDuzSol [0,KolDist*v,0]
Move KirDuzSag [0,KolDist*v,0]
$KirisDuzSol*.baseobject = $KirisDuzSol*[1].baseobject
$KirisDuzSag*.baseobject = $KirisDuzSag*[1].baseobject
$KirisDuz*.material = meditMaterials[2]
)
if SistemGorunus.chbox_duvar.checked then
(
for y = 0 to (2*KolSay)-5 do--Yan paneller - Ortadakiler (instance olarak)
(
PanelSol = box width:10 length:((KolDist/2)-2) height:KirAltH pos:[(-KolEni/2)-5, (KolDist/2)+(KolDist/4), 0] name: "PanelSol" Wirecolor:(color 100 150 200)
PanelSag = instance PanelSol name: "PanelSag" Wirecolor:(color 100 150 200)
Move PanelSag [(Sisgen+KolEni+10), 0, 0]
Move PanelSol [0,(KolDist/2)*y,0]
Move PanelSag [0,(KolDist/2)*y,0]
$PanelSol*.baseobject = $PanelSol*[1].baseobject
$PanelSag*.baseobject = $PanelSag*[1].baseobject
$Panel*.material = meditMaterials[8]
)
for z = 0 to 1 do--Yan paneller - Bastakiler (instance olarak)
(
PanelSol1 = box width:10 length:((KolDist/2)-2+(KolKal/2)+1) height:KirAltH pos:[(-KolEni/2)-5, (KolDist/4)-(KolKal/4)-0.5, 0] name: "PanelSol1" Wirecolor:(color 100 150 200)
PanelSag1 = instance PanelSol1 name: "PanelSag1" Wirecolor:(color 100 150 200)
Move PanelSag1 [(Sisgen+KolEni+10), 0, 0]
Move PanelSol1 [0,(SisUz-(KolDist/2)+8.5)*z,0]
Move PanelSag1 [0,(SisUz-(KolDist/2)+8.5)*z,0]
$PanelSol1*.baseobject = $PanelSol1*[1].baseobject
$PanelSag1*.baseobject = $PanelSag1*[1].baseobject
$Panel*.material = meditMaterials[8]
))
Zemin = plane width:(SisGen*3) length:(SisUz*3) pos:[(SisGen/2), (SisUz/2), 0] name:"Zemin" wirecolor:White --Zemin olusturur
$Zemin*.material = meditMaterials[13]
if SistemGorunus.chbox_cati.checked then --Çatiyi olusturur duruma göre
Cati = box width:(SisGen+KolEni-(2*VerPrDist)) length:(SisUz+KolKal) height:20 pos:[(SisGen/2), (SisUz/2), KolYuks] name: "Cati" wirecolor:(color 150 0 0)
$Cati*.material = meditMaterials[7]
case RenderSistem.radbut_Cam_type.state of --Kameralari olusturur ve seçimini yapar
(
1: (select $CameraDis)
2: (select $CameraIc)
)
max vpt camera
deselect objects
backgroundColor = color 255 255 255
if RenderSistem.chbox_render.checked and (RenderSistem.radbut_render_time.state == 2) then --Render seçimini yapar ve render alir
max quick render
if (RenderSistem.radbut_render_time.state == 1) then --Render alir ama önce sorar
if queryBox "Sonra hatirlat demistin. Render alayim mi?" beep:false then
max quick render
) --If-Else kapama
) --If-Else kapama
) --On-do kapama
on buttGeri pressed do
(
addRollout bilgiParalel PSS
addRollout ParalelKafesKiris PSS
removeRollout bilgiParalelWarrenM PSS
removeRollout SisParaPlWarrenM PSS
removeRollout SistemGorunus PSS
removeRollout RenderSistem PSS
removeRollout Exporting PSS
)
) --Rollout kapama--addRollout bilgiParalelWarrenM PSS
--addRollout SisParaPlWarrenM PSS
--addRollout SistemGorunus PSS
```

```
--addRollout RenderSistem PSS
```
ParalelWARREN_Simple.ms
```
--PSS = newRolloutFloater "P.S.S. 1.01 Beta" 240 770 800 100
GetFilePath = getThisScriptFilename()
SetFilePath= getFileNamePath GetFilePath

rollout bilgiParalelWarrenS "Kisa Bilgi"  RolledUp:true
(
group "Script Kullanimi"
(
label lab1 "--Asagidaki paralel baslikli WARREN-Simple" align:#left
label lab2 "      kafes kirisi sistem özelliklerinizi seçiniz" align:#left
label lab3 "      ve render - malzeme ayarlarini yapiniz." align:#left
)) rollout SisParaPlWarrenS "Sistem Parametreleri"
(
button buttGeri "<<< Üst Menüye Geri Dön <<<" border:true width:220 height:30 align:#center toolTip:"Bir Önceki Menüye Geri Döner"
group "Açikliklar"
(
spinner SistemGenis "Sistem Genisligi (cm):" range:[500,5000,2000] type:#float toolTip:"Sistemin Geçecegi Mesafeyi Belirler. (5m-50m)"
spinner SistemUzun "Sistem Uzunlugu (cm):" range:[500,30000,3000] type:#float  toolTip:"Yapinin Toplam Uzunlugunu Belirler. (5m-300m)"
)
group "Kolonlar"
(
spinner KolonYuks "Kolon Yüksekligi (cm):" range:[400,2000,800] type:#float toolTip:"Sistemin Yerden En Yüksek Noktasina Olan Mesafeyi Belirler. (4m-20m)"
spinner KolonSayisi "Kolon Sayisi:" range:[2,33,6] type:#integer toolTip:"Sistem Uzunlugu Yönündeki Toplam Aks Sayisini Belirler. (2-33)"
)
group "Diger Anlik Parametrik Bilgiler"
(
label labX1 "Kiris Yüksekligi: 125.0 cm" align:#left
label labX2 "Kirisin Yerden Yüksekligi: 675.0 cm" align:#left
label labX3 "Kolonlar Arasi Mesafe: 600.0 cm" align:#left
)
fn Paraminfo =
(
labX1.text = "Kiris Yüksekligi: " + (SistemGenis.value/16) as string + " cm"
labX2.text = "Kirisin Yerden Yüksekligi: " + (KolonYuks.value-(SistemGenis.value/16)) as string + " cm"
labX3.text = "Kolonlar Arasi Mesafe: " + ((SistemUzun.value)/(KolonSayisi.value-1)) as string + " cm"
)
on SistemGenis changed value do  Paraminfo()
on SistemUzun changed value do Paraminfo()
on KolonYuks changed value do Paraminfo()
on KolonSayisi changed value do Paraminfo()
button sil "Sil" border:true width:135 height:20 align:#left offset:[-10,0] across:2 toolTip:"Sahnedeki Bütün Nesneleri Siler."
on sil pressed do delete objects
button resett "Reset" border:true width:80 height:20 align:#right offset:[10,0] toolTip:"Sahneye Reset Atar (Yeniden Baslatir)."
on resett pressed do resetMaxFile()
button Olustur "Olustur" border:true width:220 height:30 align:#center offset:[0,0] toolTip:"Belirlediginiz Parametreler Dahilinde Sisteminizi Olusturur."
on Olustur pressed do
(
Global SisGen=SistemGenis.value--Sistem genisligi (Kolon merkezlerinden)
Global SisUz=SistemUzun.value--Sistem uzunlugu (Kolon merkezlerinden)
Global KolDist=SisUz/(KolonSayisi.value-1)--Kolonlar arasi mesafe (merkezlerinden)
Global KolSay=KolonSayisi.value--Sistemin tek siradaki kolon sayisi
Global KolEni=SisGen/50--Kolon eni
Global KolKal=KolDist/40--Kolon kalinligi
Global KolYuks=KolonYuks.value--Kolon yüksekligi
Global KirYuks=SisGen/16--Kiris yüksekligi
Global KirGen=SisGen-KolEni--Kiris genisligi
Global KirAltH=KolYuks-KirYuks--Kirisin yerden yüksekligi
Global KirHoPrYuk=SisGen/200--Kiris yatay profil yüksekliklerі(kalinliklari)
Global KirVePrEni=KirHoPrYuk/2--Kiris düsey profil enleri(kalinliklari)
Global KirVePrYuk=KirYuks-(2*KirHoPrYuk)--Kiris düsey profil yüksekliklerі
Global VerPrSayKS=(Pow (Pow SisGen 2) 0.2)/(Pow (Pow 1000 2) 0.2)--Kiris düsey profil sayisi KatSayisi (Sistem genisligi disinda, profillerin sikligini arttirip azaltmak için kullanilan bir parametre..)
Global VerPrSay=((((KirGen/2)-(KirHoPrYuk+(KirVePrEni/2)))/KirVePrYuk)*VerPrSayKS)-1--Kiris düsey profil sayisi nin yarisi
Global VerPrSay = (VerPrSay as integer)--Sayiyi tam sayi yaptik...
Global VerPrDist=((KirGen/2)-(KirHoPrYuk+(KirVePrEni/2))-(VerPrSay*KirVePrEni))/(VerPrSay+1)--Kiris düsey profiller arasi mesafe
Global VerPrDist2=((KirGen/2)-(KirHoPrYuk))/(VerPrSay+1)--Kiris düsey profiller arasi mesafe (Warren Tipleri için)
Global DiagPrUz=sqrt ((Pow KirVePrYuk 2) + (Pow VerPrDist 2))--Kiris diagonal profil uzunlugu
Global DiagPrXPo=(DiagPrUz*(KirVePrEni/2))/KirVePrYuk--Kiris diagonal profil köseden X yönündeki mesafesi
Global DiagPrYpo=KirVePrYuk*DiagPrXPo/VerPrDist--Kiris diagonal profil köseden Y yönündeki mesafesi
Global theModi = extrude amount:(KolKal/4)
if (KolDist) > 1000 or (KolDist) < 300 then messagebox "DIKKAT!!! Girilen parametler uygun degil. Parametrelerinizi gözden geçiriniz... (Kolonlar arasi mesafe 300cm ile 1000cm arasinda olmali)"
else
(
if (KolYuks-KirYuks) < 300 then messagebox "DIKKAT!!! Girilen parametler uygun degildir. Parametrelerinizi gözden geçiriniz... (Kiris yerden yüksekligi minumum 300cm olmali)"
else
(
(
CamDis = Targetcamera fov:40 nearclip:1 farclip:1000 nearrange:0 farrange:1000 mpassEnabled:off mpassRenderPerPass:off name:"CameraDis"
$CameraDis.target = Targetobject transform:(matrix3 [1,0,0] [0,1,0] [0,0,1] [SisGen/2, SisUz/10, KolYuks/2])
$CameraDis.pos = [4*SisGen/3, -(2*((13*SisGen/5))/3), 200]
CamIc = Targetcamera fov:65 nearclip:1 farclip:1000 nearrange:0 farrange:1000 mpassEnabled:off mpassRenderPerPass:off name:"CameraIc"
$CameraIc.target = Targetobject transform:(matrix3 [1,0,0] [0,1,0] [0,0,1] [SisGen/2, 4*SisUz/5, KolYuks/2])
$CameraIc.pos = [5*SisGen/6, -500, 200]
)
case RenderSistem.radbut_RenderType.state of  --Render ve malzeme tipini seçer
(
1: (renderpresets.LoadAll 0 (SetFilePath + "Standard.rps")
--Steel1
meditMaterials[1] = Standardmaterial ()
meditMaterials[1].name = "Steel1"
meditMaterials[1].adLock = off
meditMaterials[1].ambient = color 80 80 80
meditMaterials[1].Diffuse = color 110 120 130
meditMaterials[1].specularLevel = 70
meditMaterials[1].glossiness = 50
--Steel2
meditMaterials[2] = Standardmaterial ()
meditMaterials[2].name = "Steel2"
meditMaterials[2].adLock = off
meditMaterials[2].ambient = color 80 80 80
meditMaterials[2].Diffuse = color 165 170 180
meditMaterials[2].specularLevel = 70
meditMaterials[2].glossiness = 50
--Roof
meditMaterials[7] = Standardmaterial ()
meditMaterials[7].name = "Roof"
meditMaterials[7].adLock = off
meditMaterials[7].ambient = color 80 80 80
meditMaterials[7].Diffuse = color 100 0 0
meditMaterials[7].specularLevel = 55
meditMaterials[7].glossiness = 30
--Panels
meditMaterials[8] = Standardmaterial ()
meditMaterials[8].name = "Panel"
meditMaterials[8].adLock = off
meditMaterials[8].ambient = color 80 80 80
meditMaterials[8].Diffuse = color 200 200 200
meditMaterials[8].specularLevel = 55
meditMaterials[8].glossiness = 30
--Ground
meditMaterials[13] = MatteShadow ()
meditMaterials[13].name = "Ground"
meditMaterials[13].opaqueAlpha = on
meditMaterials[13].applyAtmosphere = on
--BackGround için Vray
meditMaterials[14] = Standardmaterial ()
meditMaterials[14].name = "BackGroundVR"
meditMaterials[14].diffuseMap = Gradient ()
meditMaterials[14].diffuseMap.coordinates.mappingType = 1
meditMaterials[14].diffuseMap.coordinates.mapping = 3
meditMaterials[14].diffuseMap.name = "BG_Gradient"
meditMaterials[14].diffuseMap.color1 = color 0 72 255
meditMaterials[14].diffuseMap.color2 = color 201 223 223
meditMaterials[14].diffuseMap.color3 = color 255 254 237
meditMaterials[14].diffuseMap.map1 = Noise ()
meditMaterials[14].diffuseMap.map1.name = "BG-GR_Noise"
meditMaterials[14].diffuseMap.map1.color1 = color 0 62 154
meditMaterials[14].diffuseMap.map1.color2 = color 255 244 201
meditMaterials[14].diffuseMap.map1.size = 18
meditMaterials[14].diffuseMap.map1.phase = 7
--Environment Map ( BackGroundun diffuse unu kullaniyor)
environmentMap = Gradient ()
environmentMap = meditMaterials[14].diffuseMap
)
2: (renderpresets.LoadAll 0 (SetFilePath + "Vray.rps")
--Steel1
meditMaterials[1] = VRayMtl ()
meditMaterials[1].name = "Steel1"
meditMaterials[1].Diffuse = color 110 120 130
meditMaterials[1].Reflection = color 20 20 20
meditMaterials[1].reflection_lockGlossiness = off
meditMaterials[1].hilight_glossiness = 0.9
meditMaterials[1].reflection_glossiness = 0.9
--Steel2
meditMaterials[2] = VRayMtl ()
meditMaterials[2].name = "Steel2"
meditMaterials[2].Diffuse = color 165 170 180
meditMaterials[2].Reflection = color 20 20 20
```

```
meditMaterials[2].reflection_lockGlossiness = off
meditMaterials[2].hilight_glossiness = 0.9
meditMaterials[2].reflection_glossiness = 0.9
--Roof
meditMaterials[7] = VRayMtl ()
meditMaterials[7].name = "Roof"
meditMaterials[7].Diffuse = color 100 0 0
meditMaterials[7].Reflection = color 20 20 20
meditMaterials[7].reflection_lockGlossiness = off
meditMaterials[7].hilight_glossiness = 0.65
meditMaterials[7].reflection_glossiness = 0.9
--Panels
meditMaterials[8] = VRayMtl ()
meditMaterials[8].name = "Panel"
meditMaterials[8].Diffuse = color 200 200 200
meditMaterials[8].Reflection = color 20 20 20
meditMaterials[8].reflection_lockGlossiness = off
meditMaterials[8].hilight_glossiness = 0.65
meditMaterials[8].reflection_glossiness = 0.9
--Ground
meditMaterials[13] = VRayMtlWrapper ()
meditMaterials[13].name = "Ground"
meditMaterials[13].matteSurface = on
meditMaterials[13].matte_shadows = on
meditMaterials[13].matte_shadowsAffectAlpha = on
meditMaterials[13].baseMtl = VRayMtl ()
meditMaterials[13].baseMtl.name = "GroundVR"
meditMaterials[13].baseMtl.Diffuse = color 255 255 255
--BackGround için Vray
meditMaterials[14] = VRayMtl ()
meditMaterials[14].texmap_diffuse = Gradient ()
meditMaterials[14].texmap_diffuse.coordinates.mappingType = 1
meditMaterials[14].texmap_diffuse.coordinates.mapping = 3
meditMaterials[14].texmap_diffuse.color1 = color 0 72 255
meditMaterials[14].texmap_diffuse.color2 = color 201 223 223
meditMaterials[14].texmap_diffuse.color3 = color 255 254 237
meditMaterials[14].texmap_diffuse.noiseSize = 0
meditMaterials[14].texmap_diffuse.map1 = Noise ()
meditMaterials[14].texmap_diffuse.map1.color1 = color 0 62 154
meditMaterials[14].texmap_diffuse.map1.color2 = color 255 244 201
meditMaterials[14].texmap_diffuse.map1.size = 18
meditMaterials[14].texmap_diffuse.map1.phase = 7
meditMaterials[14].name = "BackGroundVR"
meditMaterials[14].texmap_diffuse.name = "BG_Gradient"
meditMaterials[14].texmap_diffuse.map1.name = "BG-GR_Noise"
--Environment Map ( BackGroundun diffuse unu kullaniyor)
environmentMap = Gradient ()
environmentMap = meditMaterials[14].texmap_diffuse
))
fn KirSist =
(
--Kolonlarin olusturulmasi
KolonSol = box width:KolEni length:KolKal height:KolYuks name: "KolonSol" wirecolor:gray
KolonSag = instance KolonSol name:"KolonSag" wirecolor:gray
Move KolonSag [SisGen, 0, 0]
--Kiris Olusturma
KirAlt = box width:KirGen length:(KolKal/2) height:KirHoPrYuk name: "KirisAltProfil" wirecolor:orange --Alt baslik
Move KirAlt [(SisGen/2), 0, KirAltH]
KirUst = instance KirAlt name: "KirisUstProfil" wirecolor:orange --Üst baslik
move KirUst [0, 0, (KirYuks-KirHoPrYuk)]
KirPrSol = box width:KirHoPrYuk length:(KolKal/2) height:KirVePrYuk name: "KirisSolProfil" wirecolor:orange --Baslik sol profil
Move KirPrSol [(KolEni/2)+(KirHoPrYuk/2), 0, KirAltH+KirHoPrYuk]
KirPrSag = instance KirPrSol name: "KirisSagProfil" wirecolor:orange --Baslik sag profil
Move KirPrSag [KirGen-KirHoPrYuk, 0, 0]
for m = 0 to VerPrSay do
(
DiagPrA = SplineShape pos:[0,0,0] name: "KirisDiagonalProfilA" --Diagonal Profiller
addNewSpline DiagPrA
addKnot DiagPrA 1 #corner #line [0,0,0]
addKnot DiagPrA 1 #corner #line [DiagPrXPo,0,0]
addKnot DiagPrA 1 #corner #line [VerPrDist2,(KirVePrYuk-DiagPrYPo),0]
addKnot DiagPrA 1 #corner #line [VerPrDist2,KirVePrYuk,0]
addKnot DiagPrA 1 #corner #line [(VerPrDist2-DiagPrXPo),KirVePrYuk,0]
addKnot DiagPrA 1 #corner #line [0,DiagPrYPo,0]
Close DiagPrA 1
DiagPrA.wirecolor = color 255 255 255
rotate DiagPrA (eulerangles 90 0 0)
updateShape DiagPrA
Move DiagPrA [(KolEni/2)+KirHoPrYuk, (KolKal/8), KirAltH+KirHoPrYuk]
Move DiagPrA [(2*(VerprDist2))*m,0,0]
addmodifier DiagPrA theModi
)
for n = 0 to VerPrSay do
(
DiagPrB = SplineShape pos:[0,0,0] name: "KirisDiagonalProfilB" --Diagonal Profiller
addNewSpline DiagPrB
addKnot DiagPrB 1 #corner #line [0,0,0]
addKnot DiagPrB 1 #corner #line [-DiagPrXPo,0,0]
addKnot DiagPrB 1 #corner #line [-VerPrDist2,(KirVePrYuk-DiagPrYPo),0]
addKnot DiagPrB 1 #corner #line [-VerPrDist2,KirVePrYuk,0]
addKnot DiagPrB 1 #corner #line [-(VerPrDist2-DiagPrXPo),KirVePrYuk,0]
addKnot DiagPrB 1 #corner #line [0,DiagPrYPo,0]
Close DiagPrB 1
DiagPrB.wirecolor = color 255 255 255
rotate DiagPrB (eulerangles 90 0 0)
updateShape DiagPrB
Move DiagPrB [(KolEni/2)+KirHoPrYuk+(2*VerPrDist2), (KolKal/8), KirAltH+KirHoPrYuk]
Move DiagPrB [(2*(VerprDist2))*n,0,0]
addmodifier DiagPrB theModi
)
) --Fn KirSist Kapama
for t = 0 to KolSay-1 do--Kiris-Kolon sistemimizi burada tamamliyoruz ve tüm akslara kopyaliyoruz (instance olarak)
(
KirSist()
$KirisDiagonalProfilA*.baseobject = $KirisDiagonalProfilA*[1].baseobject
$KirisDiagonalProfilB*.baseobject = $KirisDiagonalProfilB*[1].baseobject
$KolonSol*.baseobject = $KolonSol*[1].baseobject
$KolonSag*.baseobject = $KolonSag*[1].baseobject
$KirisAltProfil*.baseobject = $KirisAltProfil*[1].baseobject
$KirisUstProfil*.baseobject = $KirisUstProfil*[1].baseobject
$KirisSolProfil*.baseobject = $KirisSolProfil*[1].baseobject
$KirisSagProfil*.baseobject = $KirisSagProfil*[1].baseobject
$KolonS*.material = meditMaterials[1]
$Kiris*.material = meditMaterials[2]
KirisSistem = group geometry name:"KirisSistem" select:false
Move KirisSistem [0, KolDist*t, 0]
)
for v = 0 to KolSay-2 do--Kiris sistemleri arasindaki düz profil krisleri yerlestirdik (yine instance olarak)
(
KirDuzSol = box width:(KolEni/2) length:(KolDist-KolKal) height:(KolDist/16) pos:[(-KolEni/4), (KolDist/2), (KolYuks-(KolDist/16))] name: "KirisDuzSol" Wirecolor:white
KirDuzSag = instance KirDuzSol name: "KirisDuzSag" Wirecolor:white
Move KirDuzSag [(Sisgen+(KolEni/2)), 0, 0]
Move KirDuzSol [0,KolDist*v,0]
Move KirDuzSag [0,KolDist*v,0]
$KirisDuzSol*.baseobject = $KirisDuzSol*[1].baseobject
$KirisDuzSag*.baseobject = $KirisDuzSag*[1].baseobject
$KirisDuz*.material = meditMaterials[2]
)
if SistemGorunus.chbox_duvar.checked then
(
for y = 0 to (2*KolSay)-5 do--Yan paneller - Ortadakiler (instance olarak)
(
PanelSol = box width:10 length:((KolDist/2)-2) height:KolYuks pos:[(-KolEni/2)-5, (KolDist/2)+(KolDist/4), 0] name: "PanelSol" Wirecolor:(color 100 150 200)
PanelSag = instance PanelSol name: "PanelSag" Wirecolor:(color 100 150 200)
Move PanelSag [(Sisgen+KolEni+10), 0, 0]
Move PanelSol [0,(KolDist/2)*y,0]
Move PanelSag [0,(KolDist/2)*y,0]
$PanelSol*.baseobject = $PanelSol*[1].baseobject
$PanelSag*.baseobject = $PanelSag*[1].baseobject
$Panel*.material = meditMaterials[8]
)
for z = 0 to 1 do--Yan paneller - Bastakiler (instance olarak)
(
PanelSol1 = box width:10 length:((KolDist/2)-2+(KolKal/2)+1) height:KolYuks pos:[(-KolEni/2)-5, (KolDist/4)-(KolKal/4)-0.5, 0] name: "PanelSol1" Wirecolor:(color 100 150 200)
PanelSag1 = instance PanelSol1 name: "PanelSag1" Wirecolor:(color 100 150 200)
Move PanelSag1 [(Sisgen+KolEni+10), 0, 0]
Move PanelSol1 [0,(SisUz-(KolDist/2)+8.5)*z,0]
Move PanelSag1 [0,(SisUz-(KolDist/2)+8.5)*z,0]
$PanelSol1*.baseobject = $PanelSol1*[1].baseobject
$PanelSag1*.baseobject = $PanelSag1*[1].baseobject
$Panel*.material = meditMaterials[8]
))
Zemin = plane width:(SisGen*3) length:(SisUz*3) pos:[(SisGen/2), (SisUz/2), 0] name:"Zemin" wirecolor:White --Zemin olusturur
$Zemin*.material = meditMaterials[13]
if SistemGorunus.chbox_cati.checked then --Çatiyi olusturur duruma göre
Cati = box width:(SisGen+KolEni+20) length:(SisUz+KolKal) height:20 pos:[(SisGen/2), (SisUz/2), KolYuks] name: "Cati" wirecolor:(color 150 0 0)
$Cati*.material = meditMaterials[7]
case RenderSistem.radbut_Cam_type.state of --Kameralari olusturur ve seçimini yapar
(
1: (select $CameraDis)
2: (select $CameraIc)
)
max vpt camera
deselect objects
backgroundColor = color 255 255 255
if RenderSistem.chbox_render.checked and (RenderSistem.radbut_render_time.state == 2) then --Render seçimini yapar ve render alir
max quick render
```

```
if (RenderSistem.radbut_render_time.state == 1) then  --Render alir ama önce sorar
if queryBox "Sonra hatirlat demistin. Render alayim mi?" beep:false then
max quick render
)  --If-Else kapama
)  --If-Else kapama
)  --On-do kapama
on buttGeri pressed do
(
addRollout bilgiParalel PSS
addRollout ParalelKafesKiris PSS
removeRollout bilgiParalelWarrenS PSS
removeRollout SisParaPlWarrenS PSS
removeRollout SistemGorunus PSS
removeRollout RenderSistem PSS
removeRollout Exporting PSS
))  --Rollout kapama
```

UcgenHOWE.ms

```
--PSS = newRolloutFloater "P.S.S. 1.01 Beta" 240 770 800 100
--getFileNamePath "D:\Torrent\Tezzz\Tezz\Scripting\paralelhowe.ms"
--getFileNameFile()
GetFilePath = getThisScriptFilename()
SetFilePath= getFileNamePath GetFilePath
--filein "Visualization.ms"
rollout bilgiUcgenHowe "Kisa Bilgi"  RolledUp:true
(
group "Script Kullanimi"
(
label lab1 "--Asagidaki üçgen baslikli HOWE" align:#left
label lab2 "      kafes kirisi sistem özelliklerinizi seçiniz" align:#left
label lab3 "      ve render - malzeme ayarlarini yapiniz." align:#left
))
rollout SisParaUcHowe "Sistem Parametreleri"
(
button buttGeri "<<< Üst Menüye Geri Dön <<<" border:true width:220 height:30 align:#center toolTip:"Bir Önceki Menüye Geri Döner"
group "Açikliklar"
(
spinner SistemGenis "Sistem Genisligi (cm):" range:[500,3500,1500] type:#float toolTip:"Sistemin Geçecegi Mesafeyi Belirler. (5m-35m)"
spinner SistemUzun "Sistem Uzunlugu (cm):" range:[500,30000,3000] type:#float toolTip:"Yapinin Toplam Uzunlugunu Belirler. (5m-300m)"
spinner KirisEgim "Kiris Egimi (derece):" range:[10,30,20] type:#integer toolTip:"Ana Kirisin Egimini Belirler. (10^-30^)"
)
group "Kolonlar"
(
spinner KolonYuks "Sistem Yüksekligi (cm):" range:[400,2000,800] type:#float toolTip:"Sistemin Yerden En Yüksek Noktasina Olan Mesafeyi Belirler. (4m-20m)"
spinner KolonSayisi "Kolon Sayisi:" range:[2,33,6] type:#integer toolTip:"Sistem Uzunlugu Yönündeki Toplam Aks Sayisini Belirler. (2-33)"
)
group "Diger Anlik Parametrik Bilgiler"
(
label labX1 "Kiris Yüksekligi: 288.437 cm" align:#left
label labX2 "Kirisin Yerden Yüksekligi: 511.563 cm" align:#left
label labX3 "Kolonlar Arasi Mesafe: 600.0 cm" align:#left
)
fn Paraminfo =
(
labX1.text = "Kiris Yüksekligi: " + ((SistemGenis.value/150)+((tan KirisEgim.value)*(((SistemGenis.value/50)+SistemGenis.value)/2))) as string + " cm"  --KirHoPrYuk+KirUcH
labX2.text = "Kirisin Yerden Yüksekligi: " + (KolonYuks.value-((SistemGenis.value/150)+((tan KirisEgim.value)*(((SistemGenis.value/50)+SistemGenis.value)/2)))) as string + " cm"  --KolYuks-KirYuks
labX3.text = "Kolonlar Arasi Mesafe: " + ((SistemUzun.value)/(KolonSayisi.value-1)) as string + " cm"
)
on SistemGenis changed value do  Paraminfo()
on SistemUzun changed value do Paraminfo()
on KirisEgim changed value do Paraminfo()
on KolonYuks changed value do Paraminfo()
on KolonSayisi changed value do Paraminfo()
button sil "Sil" border:true width:135 height:20 align:#left offset:[-10,0] across:2 toolTip:"Sahnedeki Bütün Nesneleri Siler."
on sil pressed do delete objects
button resett "Reset" border:true width:80 height:20 align:#right offset:[10,0] toolTip:"Sahneye Reset Atar (Yeniden Baslatir)."
on resett pressed do resetMaxFile()
button Olustur "Olustur" border:true width:220 height:30 align:#center offset:[0,0] toolTip:"Belirlediginiz Parametreler Dahilinde Sisteminizi Olusturur."
on Olustur pressed do
(
Global SisGen=SistemGenis.value--Sistem genisligi (Kolon merkezlerinden)
Global SisUz=SistemUzun.value--Sistem uzunlugu (Kolon merkezlerinden)
Global KolDist=SisUz/(KolonSayisi.value-1)--Kolonlar arasi mesafe (merkezlerinden)
Global KolSay=KolonSayisi.value--Sistemin tek siradaki kolon sayisi
Global KolEni=SisGen/50--Kolon eni
Global KolKal=KolDist/40--Kolon kalinligi
Global KolYuks=KolonYuks.value--Kolon yüksekligi
Global KirGen=SisGen+KolEni--Kiris genisligi
Global KirHoPrYuk=SisGen/150--Kiris yatay profil yüksekliklerileri(kalinliklari)
Global KirEg=KirisEgim.value--Kiristeki egim derecesi...
Global KirUcH = (tan KirEg)*(Kirgen/2)--Kiris yuksekligi (mahya H)
Global KirUstYPo = KirHoPrYuk / (cos KirEg)--Kiris profilinin yere dik kalinligi (mahyadaki)
Global KirUstXPo = KirHoPrYuk / (sin KirEg)--Kiris profilinin yere paralel uzunlugu (alt baslik ile birlestigi yerdeki)
Global KirYuks=KirHoPrYuk+KirUcH--Kiris yüksekligi
Global KirAltH=KolYuks-KirYuks--Kirisin yerden yüksekligi
Global KirVePrEni=KirHoPrYuk/2--Kiris düsey profil enleri(kalinliklari)
Global KirVePrYuk=KirYuks-(2*KirHoPrYuk)--Kiris düsey profil yüksekliklerileri
Global VerPrSayKS=(Pow (Pow SisGen 2) 0.2)/(Pow (Pow 1000 2) 0.2)--Kiris düsey profil sayisi KatSayisi (Sistem genisligi disinda, profillerin sikligini arttirip azaltmak için kullanilan bir parametre..)
Global VerPrSay=((((((KirGen-(2*KirUstXPo))/2)-(KirVePrEni/2))/KirVePrYuk)*VerPrSayKS)-1)*(tan KirEg* 6)--Kiris düsey profil sayisi nin yarisi
Global VerPrSay = (VerPrSay as integer)--Sayiyi tam sayi yaptik...
Global VerPrDist=((((KirGen-(2*KirUstXPo))/2)-(KirVePrEni/2))-(VerPrSay*KirVePrEni))/(VerPrSay+1)--Kiris düsey profiller arasi mesafe
Global DiagPrUz=sqrt ((Pow KirVePrYuk 2) + (Pow VerPrDist 2))--Kiris diagonal profil uzunlugu
Global DiagPrXPo=(DiagPrUz*(KirVePrEni/2))/KirVePrYuk--Kiris diagonal profil köseden X yönündeki mesafesi
Global DiagPrYpo=KirVePrYuk*DiagPrXPo/VerPrDist--Kiris diagonal profil köseden Y yönündeki mesafesi
Global theModi = extrude amount:(KolKal/4)
Global theModi2 = extrude amount:(KolKal/2)
if (KolDist) > 1000 or (KolDist) < 300 then messagebox "DIKKAT!!! Girilen parametler uygun degil. Parametrelerinizi gözden geçiriniz... (Kolonlar arasi mesafe 300cm ile 1000cm arasinda olmali)"
else
(
if (KolYuks-KirYuks) < 300 then messagebox "DIKKAT!!! Girilen parametler uygun degildir. Parametrelerinizi gözden geçiriniz... (Kiris yerden yüksekligi minumum 300cm olmali)"
else
(
--fn KameraYap = --Kamera Olusturucu
(
CamDis = Targetcamera fov:40 nearclip:1 farclip:1000 nearrange:0 farrange:1000 mpassEnabled:off mpassRenderPerPass:off name:"CameraDis"
$CameraDis.target = Targetobject transform:(matrix3 [1,0,0] [0,1,0] [0,0,1] [SisGen/2, SisUz/10, KolYuks/2])
$CameraDis.pos = [4*SisGen/3, -(2*((13*SisGen/5))/3), 200]
CamIc = Targetcamera fov:65 nearclip:1 farclip:1000 nearrange:0 farrange:1000 mpassEnabled:off mpassRenderPerPass:off name:"CameraIc"
$CameraIc.target = Targetobject transform:(matrix3 [1,0,0] [0,1,0] [0,0,1] [SisGen/2, 4*SisUz/5, KolYuks/2])
$CameraIc.pos = [5*SisGen/6, -500, 200]
)
case RenderSistem.radbut_RenderType.state of  --Render ve malzeme tipini seçer
(
1: (renderpresets.LoadAll 0 (SetFilePath + "Standard.rps")
--Steel1
meditMaterials[1] = Standardmaterial ()
meditMaterials[1].name = "Steel1"
meditMaterials[1].adLock = off
meditMaterials[1].ambient = color 80 80 80
meditMaterials[1].Diffuse = color 110 120 130
meditMaterials[1].specularLevel = 70
meditMaterials[1].glossiness = 50
--Steel2
meditMaterials[2] = Standardmaterial ()
meditMaterials[2].name = "Steel2"
meditMaterials[2].adLock = off
meditMaterials[2].ambient = color 80 80 80
meditMaterials[2].Diffuse = color 165 170 180
meditMaterials[2].specularLevel = 70
meditMaterials[2].glossiness = 50
--Roof
meditMaterials[7] = Standardmaterial ()
meditMaterials[7].name = "Roof"
meditMaterials[7].adLock = off
meditMaterials[7].ambient = color 80 80 80
meditMaterials[7].Diffuse = color 100 0 0
meditMaterials[7].specularLevel = 55
meditMaterials[7].glossiness = 30
--Panels
meditMaterials[8] = Standardmaterial ()
meditMaterials[8].name = "Panel"
meditMaterials[8].adLock = off
meditMaterials[8].ambient = color 80 80 80
meditMaterials[8].Diffuse = color 200 200 200
meditMaterials[8].specularLevel = 55
meditMaterials[8].glossiness = 30
--Ground
meditMaterials[13] = MatteShadow ()
meditMaterials[13].name = "Ground"
meditMaterials[13].opaqueAlpha = on
meditMaterials[13].applyAtmosphere = on
--BackGround için Vray
meditMaterials[14] = Standardmaterial ()
```

```
meditMaterials[14].name = "BackGroundVR"
meditMaterials[14].diffuseMap = Gradient ()
meditMaterials[14].diffuseMap.coordinates.mappingType = 1
meditMaterials[14].diffuseMap.coordinates.mapping = 3
meditMaterials[14].diffuseMap.name = "BG_Gradient"
meditMaterials[14].diffuseMap.color1 = color 0 72 255
meditMaterials[14].diffuseMap.color2 = color 201 223 223
meditMaterials[14].diffuseMap.color3 = color 255 254 237
meditMaterials[14].diffuseMap.map1 = Noise ()
meditMaterials[14].diffuseMap.map1.name = "BG-GR_Noise"
meditMaterials[14].diffuseMap.map1.color1 = color 0 62 154
meditMaterials[14].diffuseMap.map1.color2 = color 255 244 201
meditMaterials[14].diffuseMap.map1.size = 18
meditMaterials[14].diffuseMap.map1.phase = 7
--Environment Map ( BackGroundun diffuse unu kullaniyor)
environmentMap = Gradient ()
environmentMap = meditMaterials[14].diffuseMap
)
2: (renderpresets.LoadAll 0 (SetFilePath + "Vray.rps")
--Steel1
meditMaterials[1] = VRayMtl ()
meditMaterials[1].name = "Steel1"
meditMaterials[1].Diffuse = color 110 120 130
meditMaterials[1].Reflection = color 20 20 20
meditMaterials[1].reflection_lockGlossiness = off
meditMaterials[1].hilight_glossiness = 0.9
meditMaterials[1].reflection_glossiness = 0.9
--Steel2
meditMaterials[2] = VRayMtl ()
meditMaterials[2].name = "Steel2"
meditMaterials[2].Diffuse = color 165 170 180
meditMaterials[2].Reflection = color 20 20 20
meditMaterials[2].reflection_lockGlossiness = off
meditMaterials[2].hilight_glossiness = 0.9
meditMaterials[2].reflection_glossiness = 0.9
--Roof
meditMaterials[7] = VRayMtl ()
meditMaterials[7].name = "Roof"
meditMaterials[7].Diffuse = color 100 0 0
meditMaterials[7].Reflection = color 20 20 20
meditMaterials[7].reflection_lockGlossiness = off
meditMaterials[7].hilight_glossiness = 0.65
meditMaterials[7].reflection_glossiness = 0.9
--Panels
meditMaterials[8] = VRayMtl ()
meditMaterials[8].name = "Panel"
meditMaterials[8].Diffuse = color 200 200 200
meditMaterials[8].Reflection = color 20 20 20
meditMaterials[8].reflection_lockGlossiness = off
meditMaterials[8].hilight_glossiness = 0.65
meditMaterials[8].reflection_glossiness = 0.9
--Ground
meditMaterials[13] = VRayMtlWrapper ()
meditMaterials[13].name = "Ground"
meditMaterials[13].matteSurface = on
meditMaterials[13].matte_shadows = on
meditMaterials[13].matte_shadowsAffectAlpha = on
meditMaterials[13].baseMtl = VRayMtl ()
meditMaterials[13].baseMtl.name = "GroundVR"
meditMaterials[13].baseMtl.Diffuse = color 255 255 255
--BackGround için Vray
meditMaterials[14] = VRayMtl ()
meditMaterials[14].texmap_diffuse = Gradient ()
meditMaterials[14].texmap_diffuse.coordinates.mappingType = 1
meditMaterials[14].texmap_diffuse.coordinates.mapping = 3
meditMaterials[14].texmap_diffuse.color1 = color 0 72 255
meditMaterials[14].texmap_diffuse.color2 = color 201 223 223
meditMaterials[14].texmap_diffuse.color3 = color 255 254 237
meditMaterials[14].texmap_diffuse.noiseSize = 0
meditMaterials[14].texmap_diffuse.map1 = Noise ()
meditMaterials[14].texmap_diffuse.map1.color1 = color 0 62 154
meditMaterials[14].texmap_diffuse.map1.color2 = color 255 244 201
meditMaterials[14].texmap_diffuse.map1.size = 18
meditMaterials[14].texmap_diffuse.map1.phase = 7
meditMaterials[14].name = "BackGroundVR"
meditMaterials[14].texmap_diffuse.name = "BG_Gradient"
meditMaterials[14].texmap_diffuse.map1.name = "BG-GR_Noise"
--Environment Map ( BackGroundun diffuse unu kullaniyor)
environmentMap = Gradient ()
environmentMap = meditMaterials[14].texmap_diffuse
))
fn KirSist =
(
--Kolonlarin olusturulmasi
KolonSol = box width:KolEni length:KolKal height:KirAltH name: "KolonSol" wirecolor:gray
KolonSag = instance KolonSol name:"KolonSag" wirecolor:gray
Move KolonSag [SisGen, 0, 0]
--Kiris Olusturma
KirAlt = box width:KirGen length:(KolKal/2) height:KirHoPrYuk name: "KirisAltProfil" wirecolor:orange --Alt baslik
Move KirAlt [(SisGen/2), 0, KirAltH]
KirUst1 = SplineShape pos:[0,0,0] name: "KirisUstProfilSol" wirecolor:orange --Üst baslik sol profil
addNewSpline KirUst1
addKnot KirUst1 1 #corner #line [0,0,0]
addKnot KirUst1 1 #corner #line [(KirGen/2),KirUcH,0]
addKnot KirUst1 1 #corner #line [(KirGen/2),(KirUcH-KirUstYPo),0]
addKnot KirUst1 1 #corner #line [KirUstXPo,0,0]
Close KirUst1 1
KirUst1.wirecolor = orange
rotate KirUst1 (eulerangles 90 0 0)
updateShape KirUst1
Move KirUst1 [-(KolEni/2), (KolKal/4), KirAltH+KirHoPrYuk]
addmodifier KirUst1 theModi2
KirUst2 = SplineShape pos:[0,0,0] name: "KirisUstProfilSol" wirecolor:orange --Üst baslik sag profil
addNewSpline KirUst2
addKnot KirUst2 1 #corner #line [0,0,0]
addKnot KirUst2 1 #corner #line [-(KirGen/2),KirUcH,0]
addKnot KirUst2 1 #corner #line [-(KirGen/2),(KirUcH-KirUstYPo),0]
addKnot KirUst2 1 #corner #line [-KirUstXPo,0,0]
Close KirUst2 1
KirUst2.wirecolor = orange
rotate KirUst2 (eulerangles 90 0 0)
updateShape KirUst2
Move KirUst2 [-(KolEni/2)+KirGen, (KolKal/4), KirAltH+KirHoPrYuk]
addmodifier KirUst2 theModi2
KirPrVerOrta = SplineShape pos:[0,0,0] name: "KirisVerticalProfilOrta" wirecolor:white --Vertical profil ortadaki ana dikme
addNewSpline KirPrVerOrta
addKnot KirPrVerOrta 1 #corner #line [0,0,0]
addKnot KirPrVerOrta 1 #corner #line [(KirVePrEni),0,0]
addKnot KirPrVerOrta 1 #corner #line [(KirVePrEni),(KirUcH-KirUstYPo-((tan kirEg)*(KirVePrEni/2))),0]
addKnot KirPrVerOrta 1 #corner #line [(KirVePrEni/2),(KirUcH-KirUstYPo),0]
addKnot KirPrVerOrta 1 #corner #line [0,(KirUcH-KirUstYPo-((tan kirEg)*(KirVePrEni/2))),0]
Close KirPrVerOrta 1
KirPrVerOrta.wirecolor = white
rotate KirPrVerOrta (eulerangles 90 0 0)
updateShape KirPrVerOrta
Move KirPrVerOrta [-(KolEni/2)+(KirGen/2)-(KirVePrEni/2), (KolKal/8), KirAltH+KirHoPrYuk]
addmodifier KirPrVerOrta theModi
for i = 1 to VerPrSay do
(
KirPrVerSol = SplineShape pos:[0,0,0] name: "KirisVerticalProfilSol" wirecolor:white --Vertical profiller - sol taraftakiler
addNewSpline KirPrVerSol
addKnot KirPrVerSol 1 #corner #line [0,0,0]
addKnot KirPrVerSol 1 #corner #line [(KirVePrEni),0,0]
addKnot KirPrVerSol 1 #corner #line [(KirVePrEni),((KirUcH-KirUstYPo+((tan kirEg)*(KirVePrEni/2)))-(i*((tan KirEg)*(VerPrDist+KirVePrEni)))),0]
addKnot KirPrVerSol 1 #corner #line [0,((KirUcH-KirUstYPo-((tan KirEg)*(KirVePrEni/2)))-(i*((tan KirEg)*(VerPrDist+KirVePrEni)))),0]
Close KirPrVerSol 1
KirPrVerSol.wirecolor = white
rotate KirPrVerSol (eulerangles 90 0 0)
updateShape KirPrVerSol
Move KirPrVerSol [-(KolEni/2)+(KirGen/2)-(KirVePrEni/2), (KolKal/8), KirAltH+KirHoPrYuk]
Move KirPrVerSol [-(VerprDist+KirVePrEni)*i,0,0]
addmodifier KirPrVerSol theModi
)
for j = 1 to VerPrSay do
(
KirPrVerSag = SplineShape pos:[0,0,0] name: "KirisVerticalProfilSag" wirecolor:white --Vertical profiller - sag taraftakiler
addNewSpline KirPrVerSag
addKnot KirPrVerSag 1 #corner #line [0,0,0]
addKnot KirPrVerSag 1 #corner #line [(KirVePrEni),0,0]
addKnot KirPrVerSag 1 #corner #line [(KirVePrEni),((KirUcH-KirUstYPo-((tan KirEg)*(KirVePrEni/2)))-(j*((tan KirEg)*(VerPrDist+KirVePrEni)))),0]
addKnot KirPrVerSag 1 #corner #line [0,((KirUcH-KirUstYPo+((tan kirEg)*(KirVePrEni/2)))-(j*((tan KirEg)*(VerPrDist+KirVePrEni)))),0]
Close KirPrVerSag 1
KirPrVerSag.wirecolor = white
rotate KirPrVerSag (eulerangles 90 0 0)
updateShape KirPrVerSag
Move KirPrVerSag [-(KolEni/2)+(KirGen/2)-(KirVePrEni/2), (KolKal/8), KirAltH+KirHoPrYuk]
Move KirPrVerSag [(VerprDist+KirVePrEni)*j,0,0]
addmodifier KirPrVerSag theModi
)
for m = 1 to VerPrSay do
(
EgAn = ((KirUcH-KirUstYPo+((tan KirEg)*(KirVePrEni/2)))-(m*((tan KirEg)*(VerPrDist+KirVePrEni)))) --Her yeni egim acisi için gerekli olan ve degisen profil yükseligi
aci = aTan (EgAn/VerPrDist) --Her Dikey profil arasi olusan yeni egim acisi
DiagPrYPoUc = ((KirVePrEni/2) / (Sin aci)) --Her yeni egim acisi gerekli uzaklik - X yönünde
DiagPrYPoUc = ((KirVePrEni/2) / (Cos aci)) --Her yeni egim acisi gerekli uzaklik - Y yönünde
DiagPrA = SplineShape pos:[0,0,0] name: "KirisDiagonalProfilA" --Diagonal Profiller
addNewSpline DiagPrA
```

```
addKnot DiagPrA 1 #corner #line [0,0,0]
addKnot DiagPrA 1 #corner #line [-DiagPrXPoUc,0,0]
addKnot DiagPrA 1 #corner #line [-VerPrDist,EgAn-DiagPrYPoUc,0]
addKnot DiagPrA 1 #corner #line [-VerPrDist,EgAn,0]
addKnot DiagPrA 1 #corner #line [(-VerPrDist+DiagPrXPoUc),EgAn+((tan kirEg)*DiagPrXPoUc),0] --Burda ufak bir egim hesap eksigi var sonra kontrol et...!!!!!!!!!!!
addKnot DiagPrA 1 #corner #line [0,DiagPrYPoUc,0]
Close DiagPrA 1
DiagPrA.wirecolor = color 255 255 255
rotate DiagPrA (eulerangles 90 0 0)
updateShape DiagPrA
Move DiagPrA [-(KolEni/2)+(KirGen/2)-(KirVePrEni/2)+(VerprDist+KirVePrEni), (KolKal/8), KirAltH+KirHoPrYuk]
Move DiagPrA [-(VerprDist+KirVePrEni)*m,0,0]
addmodifier DiagPrA theModi
)
for n = 1 to VerPrSay do
(
EgAn = ((KirUcH-KirUstYPo+((tan KirEg)*(KirVePrEni/2)))-(n*((tan KirEg)*(VerPrDist+KirVePrEni)))) --Her yeni egim acisi için gerekli olan ve degisen profil yükseligi
aci = aTan (EgAn/VerPrDist) --Her Dikey profil arasi olusan yeni egim acisi
DiagPrXPoUc = ((KirVePrEni/2) / (Sin aci)) --Her yeni egim acisi gerekli uzaklik - X yönünde
DiagPrYPoUc = ((KirVePrEni/2) / (Cos aci)) --Her yeni egim acisi gerekli uzaklik - Y yönünde
DiagPrB = SplineShape pos:[0,0,0] name: "KirisDiagonalProfilB" --Diagonal Profiller
addNewSpline DiagPrB
addKnot DiagPrB 1 #corner #line [0,0,0]
addKnot DiagPrB 1 #corner #line [DiagPrXPoUc,0,0]
addKnot DiagPrB 1 #corner #line [VerPrDist,EgAn-DiagPrYPoUc,0]
addKnot DiagPrB 1 #corner #line [VerPrDist,EgAn,0]
addKnot DiagPrB 1 #corner #line [(VerPrDist-DiagPrXPoUc),EgAn+((tan kirEg)*DiagPrXPoUc),0] --Burda ufak bir egim hesap eksigi var sonra kontrol et...!!!!!!!!!!!
addKnot DiagPrB 1 #corner #line [0,DiagPrYPoUc,0]
Close DiagPrB 1
DiagPrB.wirecolor = color 255 255 255
rotate DiagPrB (eulerangles 90 0 0)
updateShape DiagPrB
Move DiagPrB [-(KolEni/2)+(KirGen/2)+(KirVePrEni/2)-(VerprDist+KirVePrEni), (KolKal/8), KirAltH+KirHoPrYuk]
--Move DiagPrB [(KolEni/2)+KirHoPrYuk+((VerprDist+KirVePrEni)*(VerPrSay+2))-KirVePrEni, (KolKal/8), KirAltH+KirHoPrYuk]
Move DiagPrB [(VerprDist+KirVePrEni)*n,0,0]
addmodifier DiagPrB theModi
)) --Fn KirSist Kapama
for t = 0 to KolSay-1 do--Kiris-Kolon sistemimizi burada tamamliyoruz ve tüm akslara kopyaliyoruz (instance olarak)
(
KirSist()
$KolonSol*.baseobject = $KolonSol*[1].baseobject
$KolonSag*.baseobject = $KolonSag*[1].baseobject
$KirisAltProfil*.baseobject = $KirisAltProfil*[1].baseobject
$KolonS*.material = meditMaterials[1]
$Kiris*.material = meditMaterials[2]
KirisSistem = group geometry name:"KirisSistem" select:false
Move KirisSistem [0, KolDist*t, 0]
)
for v = 0 to KolSay-2 do--Kiris sistemleri arasindaki düz profil krisleri yerlestirdik (yine instance olarak)
(
KirDuzSol = box width:(KolEni/2) length:(KolDist-KolKal) height:(KolDist/16) pos:[(-KolEni/4), (KolDist/2), (KirAltH-(KolDist/16))] name: "KirisDuzSol" Wirecolor:white
KirDuzSag = instance KirDuzSol name: "KirisDuzSag" Wirecolor:white
Move KirDuzSag [(Sisgen+(KolEni/2)), 0, 0]
Move KirDuzSol [0,KolDist*v,0]
Move KirDuzSag [0,KolDist*v,0]
$KirisDuzSol*.baseobject = $KirisDuzSol*[1].baseobject
$KirisDuzSag*.baseobject = $KirisDuzSag*[1].baseobject
$KirisDuz*.material = meditMaterials[2]
)
if SistemGorunus.chbox_duvar.checked then
(
for y = 0 to (2*KolSay)-5 do--Yan paneller - Ortadakiler (instance olarak)
(
PanelSol = box width:10 length:((KolDist/2)-2) height:(KirAltH+KirHoPrYuk) pos:[(-KolEni/2)-5, (KolDist/2)+(KolDist/4), 0] name: "PanelSol" Wirecolor:(color 100 150 200)
PanelSag = instance PanelSol name: "PanelSag" Wirecolor:(color 100 150 200)
Move PanelSag [(Sisgen+KolEni+10), 0, 0]
Move PanelSol [0,(KolDist/2)*y,0]
Move PanelSag [0,(KolDist/2)*y,0]
$PanelSol*.baseobject = $PanelSol*[1].baseobject
$PanelSag*.baseobject = $PanelSag*[1].baseobject
$Panel*.material = meditMaterials[8]
)
for z = 0 to 1 do--Yan paneller - Bastakiler (instance olarak)
(
PanelSol1 = box width:10 length:((KolDist/2)-2+(KolKal/2)+1) height:(KirAltH+KirHoPrYuk) pos:[(-KolEni/2)-5, (KolDist/4)-(KolKal/4)-0.5, 0] name: "PanelSol1" Wirecolor:(color 100 150 200)
PanelSag1 = instance PanelSol1 name: "PanelSag1" Wirecolor:(color 100 150 200)
Move PanelSag1 [(Sisgen+KolEni+10), 0, 0]
Move PanelSol1 [0,(SisUz-(KolDist/2)+8.5)*z,0]
Move PanelSag1 [0,(SisUz-(KolDist/2)+8.5)*z,0]
$PanelSol1*.baseobject = $PanelSol1*[1].baseobject
$PanelSag1*.baseobject = $PanelSag1*[1].baseobject
$Panel*.material = meditMaterials[8]
))
Zemin = plane width:(SisGen*3) length:(SisUz*3) pos:[(SisGen/2), (SisUz/2), 0] name:"Zemin" wirecolor:White --Zemin olusturur
$Zemin*.material = meditMaterials[13]
if SistemGorunus.chbox_cati.checked then --Çatiyi olusturur duruma göre
(
CatiUc = SplineShape pos:[0,0,0] name: "Cati"
addNewSpline CatiUc
addKnot CatiUc 1 #corner #line [0,0,0]
addKnot CatiUc 1 #corner #line [KirGen/2,KirUcH,0]
addKnot CatiUc 1 #corner #line [KirGen,0,0]
addKnot CatiUc 1 #corner #line [KirGen,10,0]
addKnot CatiUc 1 #corner #line [KirGen/2,KirUcH+10,0]
addKnot CatiUc 1 #corner #line [0,10,0]
Close CatiUc 1
CatiUc.wirecolor = color 255 255 255
rotate CatiUc (eulerangles 90 0 0)
updateShape CatiUc
Move CatiUc [-(KolEni/2), -(KolKal/2), KirAltH+KirHoPrYuk]
addmodifier CatiUc (extrude amount:-(SisUz+KolKal))
)
$Cati*.material = meditMaterials[7]
case RenderSistem.radbut_Cam_type.state of --Kameralari olusturur ve seçimini yapar
(
1: (select $CameraDis)
2: (select $CameraIc)
)
max vpt camera
deselect objects
backgroundColor = color 255 255 255
if RenderSistem.chbox_render.checked and (RenderSistem.radbut_render_time.state == 2) then --Render seçimini yapar ve render alir
max quick render
if (RenderSistem.radbut_render_time.state == 1) then --Render alir ama önce sorar
if queryBox "Sonra hatirlat demistin. Render alayim mi?" beep:false then
max quick render
) --If-Else kapama
) --If-Else kapama
) --On-do kapama
on buttGeri pressed do
(
addRollout bilgiUcgen PSS
addRollout UcgenKafesKiris PSS
removeRollout bilgiUcgenHowe PSS
removeRollout SisParaUcHowe PSS
removeRollout SistemGorunus PSS
removeRollout RenderSistem PSS
removeRollout Exporting PSS
)) --Rollout kapama
```

Exporting.ms

```
rollout Exporting "Export Seçenekleri" RolledUp:true
(
group "Script Kullanimi"
(
Dropdownlist DlistExport "Format seç" items:#(".3ds", ".Fbx", ".Dwg", ".Dxf", ".Obj", ".Dae", ".Wrl") width:100 align:#left toolTip:"Oluşturmak istediginiz formatı seçin."
button Exportit "Press To Export" width:100 height:30 align:#Left across:2 toolTip:"Seçtiginiz formatta export eder."
button OpenFolder "Open Folder" width:80 height:30 align:#Right toolTip:"Export ettiginiz klasörü açar."
on Exportit pressed do
(
 exportFile ((GetDir #export) + "\PSS" + (DlistExport.selected)) #noPrompt
)
on OpenFolder pressed do
(
 shellLaunch "explorer.exe" (GetDir #export)
)
)
)
```

ÖZGEÇMİŞ

KİŞİSEL BİLGİLER

Adı Soyadı	:Engin Sedat ÖZDEMİR
Doğum Tarihi ve Yeri	:25.11.1981 / Ankara
Yabancı Dili	:İngilizce
E-posta	:enginsedat@gmail.com

ÖĞRENİM DURUMU

Derece	**Alan**	**Okul/Üniversite**	**Mezuniyet Yılı**
Y. Lisans	Mimarlık	YTÜ	2016
Lisans	Mimarlık	YTÜ	2006
Lise		Eryaman Lisesi(Ankara)	1998

İŞ TECRÜBESİ

Yıl	**Firma/Kurum**	**Görevi**
2006-2016	OZ Mimarlık	Mimar
2003-2006	SKM Mimarlık	Mimar