

**T.C.
İNÖNÜ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

**YAPI BİLGİ MODELLEMESİ TABANLI KABA METRAJ HESAPLAYAN
VE 3B GÖRSELLEŞTİREN YAZILIMIN GELİŞTİRİLMESİ**

YÜKSEK LİSANS TEZİ

**Öğrenci Adı SOYADI
Faruk ERGEN**

İnşaat Mühendisliği Anabilim Dalı

Tez Danışmanı: Doç. Dr. Önder Halis BETTEMİR

TEMMUZ 2022

T.C
İNÖNÜ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

**YAPI BİLGİ MODELLEMESİ TABANLI KABA METRAJ HESAPLAYAN
VE 3B GÖRSELLEŞTİREN YAZILIMIN GELİŞTİRİLMESİ**

YÜKSEK LİSANS TEZİ

Öğrenci Adı SOYADI
Faruk ERGEN
(36193621013)

İnşaat Mühendisliği Anabilim Dalı

Tez Danışmanı: Doç. Dr. Önder Halis BETTEMİR

TEMMUZ 2022

TEŞEKKÜR VE ÖNSÖZ

Bu tez çalışmasının her aşamasında yardım, öneri, bilgi, tecrübe ve desteklerini esirgemedi beni her konuda yönlendiren danışman hocam Sayın Doç. Dr. Önder Halis Bettemir'e,

Çalışmalarında ayrıca tüm hayatım boyunca olduğu gibi bu çalışmalarım süresince benden her türlü desteklerini esirgemeyen aileme,

teşekkür ederim.



ONUR SÖZÜ

Yüksek lisans tezi olarak sunduğum “Yapı Bilgi Modellemesi Tabanlı Kaba Metraj Hesaplayan ve 3B Görselleştiren Yazılımın Geliştirilmesi” başlıklı bu çalışmanın bilimsel ahlak ve geleneklere aykırı düşecek bir yardıma başvurmaksızın tarafımdan yazıldığına ve yararlandığım bütün kaynakların hem metin içinde hem de kaynakçada yöntemine uygun biçimde gösterilenlerden oluştuğunu belirtir, bunu onurumla doğrularım.

Faruk Ergen



İÇİNDEKİLER

TEŞEKKÜR VE ÖNSÖZ	i
ONUR SÖZÜ	ii
İÇİNDEKİLER.....	iii
ÇİZELGELER DİZİNİ.....	iv
ŞEKİLLER DİZİNİ.....	v
SEMBOLLER VE KISALTMALAR	vi
ÖZET	vii
ABSTRACT	viii
1. GİRİŞ	1
1.1 Tezin Amacı.....	1
1.2 Tezin Kapsamı	2
2. LİTERATÜR İNCELEMESİ.....	4
2.1 Bilgi ve İletişim Teknolojileri	4
2.2 Yapı Bilgi Modellemesi.....	4
2.3 Önemli YBM Yazılımları	7
2.3.1 Autodesk Revit	7
2.3.2 ArchiCAD.....	8
2.3.3 Diğer yazılımlar	8
2.4 IFC	8
2.5 YBM Verimliliği	10
2.6 YBM Sürecinde Metraj ve Maliyet Tahmini	15
2.7 YBM ve Oyun Motorları	17
2.8 Python Programlama Dili	18
3. YAZILIMIN GELİŞTİRİLMESİ	22
3.1 Çizim Ara Yüzünün Oluşturulması	23
3.2 Çizim İşleminin Gerçekleştirilmesi	26
3.3 Veri Tabanının Oluşturulması ve Yönetilmesi	31
3.4 Çizilen Elemanların Komşuluklarının Tespit Edilmesi	33
3.5 Metraj ve Maliyet Hesabı	34
3.5.1 Kalıp metrajı	34
3.5.2 Beton metrajı	34
3.5.3 İskele metrajı	35
3.5.4 Donatı Metrajı	35
3.6 Yapının 3 Boyutlu Görselleştirilmesi	45
4. VAKA ÇALIŞMALARI.....	48
4.1 Vaka Çalışması 1	48
4.2 Vaka Çalışması 2	51
4.3 Vaka Çalışması 3	55
5. TARTIŞMA.....	57
5.1 Bulguların Değerlendirilmesi	57
5.2 Sınırlılıklar	58
6. SONUÇ VE ÖNERİLER	59
KAYNAKLAR.....	60
ÖZGEÇMİŞ	64

ÇİZELGELER DİZİNİ

Çizelge 4.1: Kolon donatı metrajı.....	53
Çizelge 4.2: Kiriş donatı metrajı.....	54
Çizelge 4.3: Döşeme donatı metrajı.....	55
Çizelge 4.4: Kalıp ve Beton metrajı sonuçları (Ergen ve Bettemir, 2022)	56
Çizelge 4.5: Metraj değerlerinin yüzdelik hata değerleri (Ergen ve Bettemir, 2022).....	56



ŞEKİLLER DİZİNİ

Şekil 2.1: Industry Foundation Classes (IFC) uluslararası veri alışverişi formatının içeriği ve bir YBM yazılımında görüntülenme süreci	10
Şekil 2.2: YBM Olgunluk Seviyeleri (Building Information Modelling (BIM) Working Party Strategy Paper, 2011)	11
Şekil 3.1: Geliştirilen Yazılımın Akış Şeması	22
Şekil 3.2: Frame sınıfı kullanılarak çerçeve oluşturulması	24
Şekil 3.3: Kullanıcı Ara yüzü penceresinde butonların ve çizim ekranının konumları	24
Şekil 3.4: Kolon butonuna basıldığında yapı elemanlarından Kolona ait özelliklerin girilmesini sağlayan pencerenin açılması	25
Şekil 3.5: Label ve Entry sınıflarının kullanımı	25
Şekil 3.6: Kolon penceresinde kolonun özelliklerinin girilmesinin ardından “ÇİZ” butonuna basılması	26
Şekil 3.7: Kolonun çiziminin yapılacağı aksların kesiştiği noktaya farenin sol tuşu ile tıklanması	27
Şekil 3.8: Kolon çiziminin gerçekleştirilmesi	27
Şekil 3.9: Kiriş penceresinde kirişin özelliklerinin girilmesinin ardından “ÇİZ” butonuna basılması	28
Şekil 3.10: Kiriş çiziminin gerçekleştirilmesi ve 3B Modelinin görüntülenmesi	28
Şekil 3.11: Döşeme penceresinde döşeme özelliklerinin girilmesinin ardından “ÇİZ” butonuna basılması	29
Şekil 3.12: Döşeme çiziminin gerçekleştirilmesi ve 3B Modelinin görüntülenmesi	29
Şekil 3.13: 3 katlı A Yapısının Kalıp Planı Çizimi	31
Şekil 3.14: 2 katlı B Yapısının Kalıp Planı Çizimi	31
Şekil 3.15: sqlite3 ile oluşturulan B yapısına ait döşemelerin veri tabanı	32
Şekil 3.16: sqlite3 ile veri tabanı oluşturma	32
Şekil 3.17: S01 Kolonu ve Komşu Kirişler	33
Şekil 3.18: Birim Fiyat Penceresi	35
Şekil 3.19: Döşemeye ait verilerin girildiği pencere	38
Şekil 3.20: Döşeme donatı metrajı hesabı akış şeması	39
Şekil 3.21: İki doğrultuda çalışan döşemenin sürekli kenarı boyunca pilye donatılarının uzatılması	40
Şekil 3.22: Kiriş donatı metrajı algoritmasının akış şeması	42
Şekil 3.23: Kolon donatı metrajı hesabı akış şeması	43
Şekil 3.24: Ursina Koordinat Sistemi	45
Şekil 3.25: A yapısının dokusuz modeli	46
Şekil 3.26: B yapısının dokusuz 3B Modeli	46
Şekil 4.1: A yapısına ait Metraj ve Maliyet Hesabı Raporu	48
Şekil 4.2: B yapısına ait Metraj ve Maliyet Hesabı Raporu	49
Şekil 4.3: A yapısının beton dokulu 3B Modeli	50
Şekil 4.4: B yapısının beton dokulu 3B Modeli	50
Şekil 4.5: A yapısında gezinme işleminin gerçekleştirilmesi	50
Şekil 4.6: B yapısında gezinme işleminin gerçekleştirilmesi	51
Şekil 4.7: Donatı metrajı hesaplanan yapının Kalıp Planı	51
Şekil 4.8: Döşeme donatı planı	52
Şekil 4.9: Vaka çalışması 3 Kalıp Planı (Ergen ve Bettemir, 2022)	56

SEMBOLLER VE KISALTMALAR

YBM	: Yapı Bilgi Modeli
3B	: 3 Boyut
BDÇ	: Bilgisayar Destekli Çizim
BDT	: Bilgisayar Destekli Tasarım
BİT	: Bilgi ve İletişim Teknolojisi
BT	: Bilgi Teknolojisi
TS 500	: Betonarme Yapıların Tasarım ve Yapım Kuralları
TBDY 2018	: Türkiye Bina Deprem Yönetmeliği 2018
IFC	: Industry Foundation Classes
EPT	: Entegre Proje Teslimi
MVD	: Model Görünüm Tanımları
HVAC	: Isıtma, Havalandırma ve Soğutma (Heating, ventilation and air conditioning)
AEC	: Mimarlık, Mühendislik ve İnşaat
MT	: Metraj Tablosu
PDF	: Taşınabilir Belge Formatı
YOS	: YBM Olgunluk Seviyesi
4B	: 4 Boyut
5B	: 5 Boyut
CAD	: Bilgisayar Destekli Tasarım-Grafik
YDM	: Yaşam Döngü Maliyeti (YDM)
GUI	: Grafikselsel Kullanıcı Arayüzü (Graphical User Interface)
BMN	: Bileşik Modellenen Nesne
AMN	: Ayrık Modellenen Nesne
Ln	: Kolon net açıklığı

ÖZET

Yüksek Lisans Tezi

YAPI BİLGİ MODELLEMESİ TABANLI KABA METRAJ HESAPLAYAN VE 3B GÖRSELLEŞTİREN YAZILIMIN GELİŞTİRİLMESİ

FARUK ERGEN

İnönü Üniversitesi
Fen Bilimleri Enstitüsü
İnşaat Mühendisliği Anabilim Dalı

64+viii sayfa

2022

Danışman: Doç. Dr. Önder Halis BETTEMİR

Bir inşaat projesinde kullanılacak olan malzemelerin miktarının ve maliyetinin tahmini tasarım sürecinin en önemli aşamalarından birini oluşturmaktadır. Günümüzde Yapı Bilgi Modellemesi (YBM) sürecinde kullanılan yazılımlar ile metraj hesaplamaları ve maliyet tahmini geleneksel yöntemlere göre oldukça hızlı bir biçimde gerçekleştirilmektedir. Fakat YBM yazılımlarının yapı elemanları arasındaki komşuluk ilişkilerini tam olarak belirleyememesi ve yapı elemanlarının anlamsal modelini dikkate almadan sadece geometrik şekilleri dikkate alarak metraj hesabını gerçekleştirmesinden dolayı sonuçlar gerçek değerlere göre önemli ölçüde sapma göstermektedir. Ayrıca kullanılan yazılımların lisanslama ücretleri oldukça pahalıdır ve genellikle yerel mevzuatlara göre özelleştirilmeden kullanılmaktadır. Belirtilen nedenler uluslararası yazılımların inşaat sektöründeki uygulanabilirliğini azaltmaktadır. Bu tez çalışmasında Türk inşaat sektörünün YBM uygulaması üzerine karşılaştığı sorunları çözebilmek için yerli, ücretsiz ve açık kaynaklı bir YBM yazılımının geliştirilmesi amaçlanmıştır. Yazılım, python programlama dili kullanılarak yapının kaba metrajını yapıyı oluşturan elemanların komşuluk ilişkilerini ve anlamsal temsillerini dikkate alarak hesaplamaktadır. Ayrıca geliştirilen yazılımın yapıyı 3B görselleştirerek YBM sürecine yardımcı olması amaçlanmaktadır. Yazılımın geliştirilme yöntemi, metraj hesaplamaları için kullanılan algoritmalar ve yapının 3B görselleştirme süreci vaka çalışmaları ile test edilmiştir. Test sonuçlarında beton, kalıp, kalıp iskelesi ve donatı metraj değerlerinin hatasız hesaplandığı belirlenmiştir. Ayrıca yapının 2B çizimlerine sadece derinlik bilgisi girilerek başarılı bir şekilde 3B görselleştirmesinin yapıldığı, binadaki merdiven ve asansör boşluklarının gösterilebildiği ve bina içinde yürümenin kesintisiz biçimde gerçekleştirilebildiği tespit edilmiştir.

Anahtar Kelimeler: Yapı Bilgi Modellemesi, Python programlama dili, Metraj, 3 Boyutlu Görselleştirme

ABSTRACT

Master Thesis

DEVELOPMENT OF BUILDING INFORMATION MODELING BASED QUANTITY CALCULATION AND 3D VISUALIZATION SOFTWARE

FARUK ERGEN

Inonu University
Graduate School of Nature and Applied Sciences
Department of Civil Engineering

64+viii sayfa

2022

Supervisor: Doç. Dr. Önder Halis BETTEMİR

Estimating the amount and cost of materials to be used in a construction project is one of the most important stages of the design process. Today, with the software used in the Building Information Modeling (BIM) process, quantity calculations and cost estimation are performed much faster than traditional methods. However, the results show significant deviations from the real values, since the BIM software cannot determine the neighborhood relations between the building elements exactly and calculates only the geometric shapes without considering the semantic model of the building elements. In addition, the licensing fees of the software used are quite expensive and are generally used without customization according to local regulations. In this study, it is aimed to develop a software that helps the BIM process with a graphical user interface, which calculates the concrete, formwork, formwork scaffolding and reinforcement quantities of the building by using the free open source python programming language, taking into account the neighborhood relations and semantic representations of the elements that make up the building. The development framework of the software, the algorithms used for the quantity calculations and the 3D visualization process of the structure have been tested with case studies. In the test results, it was determined that the quantity values were calculated with minimum error and the structure was successfully visualized.

Keywords: Building Information Modeling, python programming language, quantity takeoff, 3D Visualization

1. GİRİŞ

Bilgisayar Destekli Tasarım (BDT, *CAD*) programları iş hayatına girmeden önce çizim masalarında T-cetvelleri kullanılarak kağıt üzerinde teknik çizimler yapılmaktaydı. BDT yazılımları sayesinde T-cetvelleri ile yapılan çizimler yerini bilgisarda yapılan çizimlere bıraktı. Bu sayede proje üzerinde yapılan revizeler çok daha zahmetsiz, düşük maliyetli ve hızlı yapılabilir hale gelmiştir. Çizim ofisi ile şantiye arasının uzak olduğu durumlarda elektronik belge gönderimi ile çok hızlı biçimde revize projeler ulaştırılabilir hale gelmiştir. Bilgisayar destekli çizim (BDÇ) yazılımların güncel teknolojinin sunduğu imkanlara kıyasla yetersiz kalması sonucu Yapı Bilgi Modellemesinin (YBM) geliştirilmesi ile birlikte inşaat sektörünün uyguladığı yapım yönetimi yöntemlerinde köklü değişimler başlamıştır. AutoCAD, AllPlan ve Autodesk Revit gibi yazılımların geliştirildiği YBM sürecinde 2B çizimi, 3B modeller oluşturmayı ve işbirliğini arttıran programların kullanılması ile birlikte günlerce süren çizim ve görselleştirme işlemleri günümüzde çok daha kısa sürede ve daha az çaba ile gerçekleştirilmektedir.

Yapı analiz programı ile modellenip son haline getirilen tasarım dosyası YBM yazılımlarına aktararak yapı ile ilgili maliyet ve süre analizleri yapıp hızlı bir şekilde inşaat sürecine geçilebilmektedir. Örneğin karmaşık mühendislik hesaplamaları Sap2000 gibi programlar ile çok daha kısa sürede gerçekleştirilmektedir. Tasarımın son hali dxf uzantılı dosya biçiminde başka yazılımlara aktarılabilir. Fakat aktarılan dosya her versiyonda uyumlu biçimde açılmamakta ve uzun süren insan emeğine dayalı revize işlemlerinin yapılmasını gerektirmektedir. Belirtilen soruna ilave olarak kullanılan yazılımların pahalı olması, hatalı metraj sonuçları sunması, yazılımı kullanan mühendislerin yazılımla ilgili eğitim eksikliği ve müşterilerin güvensizliği gibi risklerden dolayı önemli avantajlar sunmasına rağmen YBM'nin kullanımı kısıtlıdır. YBM'nin kullanımını artırmak için bahsedilen sorunların ortadan kaldırılması gereklidir. Bu çalışmada YBM kullanımının önündeki engellerden biri olan pahalı yazılım lisansları ve hatalı metraj hesabı risklerini ortadan kaldırmak için ücretsiz açık kaynaklı programlama dilleri olan python ve sql kullanılarak daha az hata ile metraj hesabı yapan bir program geliştirilmiştir.

1.1 Tezin Amacı

Tamamen ücretsiz ve açık kaynaklı olan Python programlama dili, SQLite tabanlı veri tabanı motoru Sqlite3 ve Ursina oyun motoru kullanılarak bir yapının tasarım ve

yapım sürecinin otomasyonunu ve 3B görselleştirilmesini sağlayan YBM tabanlı bir program geliştirerek verimliliği artırmayı; yüksek doğrulukta metraj hesaplamayı, tasarım hatalarını en aza indirerek maliyeti azaltıp kaynak tasarrufu yapmayı, inşaat aşamalarının 3 boyutlu görselleştirilerek inşaat sürecinin 4 boyutlu simülasyonunun yapılması amaçlanmaktadır.

Tez çalışması sonucu erişilmek istenen hedefler;

1. Python programlama dilinde çizim arayüzü geliştirerek yapının yapısal elemanlarının ile boyutlarının geliştirilecek yazılıma tanıtılmasını sağlayacak arayüzü geliştirmek.
2. Yazılıma eklenecek veritabanının ücretsiz açık kaynaklı veri tabanı motoru olan SQLite ile oluşturulması ve en az 20 adet konut amaçlı binalarda kullanılacak yapısal elemanın yapım tarifini ekleyip çizim yoluyla tanımlaması yapılan elemanın öz niteliklerini veritabanından seçip atayabilen veritabanını oluşturmak.
3. Yapının boyutları ile yapısal elemanların komşuluklarını göz önüne alarak elemanlar arasındaki komşulukları ve temas eden yüzeyleri belirleyip minhaları hesaplayabilen ve metraj hesaplamalarını sistematik biçimde gerçekleştiren bir uygulama geliştirmek.
4. İnşaatın 3 Boyutlu görselleştirilmesini python programlama dili ile kodlanmış olan ücretsiz açık kaynaklı Ursina oyun motoru (aynı zamanda bir kütüphane) ile gerçekleştirmek ve kullanıcının 3B model vasıtasıyla yapı içerisinde gezinebilmesini sağlayan uygulamayı geliştirmek.
5. Mühendislik alanında sınırlı kaynaklarla YBM tabanlı yazılımın geliştirilebileceğinin ve pahalı lisans ücretlerine sahip yazılımların alternatifinin oluşturulabileceğinin farkındalığını oluşturmak.

1.2 Tezin Kapsamı

YBM tabanlı inşaat planlaması ve yapının 3 boyutlu görselleştirilmesi ile simülasyonunu gerçekleştiren programın algoritmasının oluşturulup kodlanması bu tezin konusunu oluşturmaktadır. İnşaat mühendislerine ve müteahhitlere yönelik oluşturulacak olan bu program metraj, maliyet analizi, iş programı oluşturma, iş ilerleme raporlarını

değerlendirme, iş programını revize etme ve yapının 3B görselleştirilip yapı içerisinde gezinebilme iş paketlerini kapsamaktadır.

Tez çalışmasının konusunu oluşturan yazılımı inşaat yönetimini gerçekleştiren diğer yazılımlardan ayıran en önemli özellikleri kullanıcıya oyun motoru kullanılarak 3B yapı içerisinde gezinebilme imkanı sağlaması ve yazılım sonucu beton, kalıp ve donatı metrajı değerlerindeki hata oranının en aza indirilmesi ile verimliliği artırmasıdır. Tez çalışması için YBM, inşaat yönetimi, veri tabanı, simülasyon konuları detaylı olarak irdelenmiş olup kapsamlı bir literatür taraması yapılmıştır.

Tezin ilerleyen kısımları aşağıdaki gibi belirtilmiştir. Bölüm 2’de YBM sürecinin inşaat sektöründeki kullanımı, inşaat profesyonelleri arasındaki yaygınlığı ve bir projenin tasarım ve inşaat aşamalarında verimliliği artırmadaki etkisi incelenmiştir.

Bölüm 3’te tez çalışmasındaki yazılımın grafiksel kullanıcı arayüzünü (GUI) kodlamak için kullanılan Python programlama dili ve kütüphaneleri, veri tabanı işlemleri için kullanılan SQLite veritabanı motoru ve yapının 3B görselleştirilmesi, simülasyonu ve yapı içerisinde kullanıcının gezinebilmesi için kullanılan Ursina 4.0 oyun motoru hakkında bilgi verilmiştir ve tezin konusunu oluşturan yazılımın geliştirilme aşamaları vaka çalışmaları ile örneklendirilerek anlatılmıştır. Bölüm 4’te vaka çalışmaları sonucunda elde edilen bulgular değerlendirilmiştir. Ayrıca tez çalışmasında geliştirilen yazılımın geliştirilme sürecindeki engeller sunulmuş olup gelecekteki çalışmalar için geliştirilen yazılımın üzerine önerilerde bulunulmuştur. Bölüm 5’te elde edilen sonuçlar değerlendirilmiştir.

2. LİTERATÜR İNCELEMESİ

2.1 Bilgi ve İletişim Teknolojileri

Haberleşme teknolojisini kullanarak bilgiye erişimi sağlayan yöntemler Bilgi ve İletişim Teknolojileri (BİT) olarak tanımlanır. Bilgi Teknolojisinin (BT) genişletilmiş halidir. BİT ürün imalatı ve hizmet sektörlerinde birçok sorumluluğu üstlenmektedir. Ayrıca toplumsal ve ticari alanlarda kişi ve kurumların birbirleri ile ilişkilerini sürdürmelerine yardımcı olur.

BİT, diğer endüstrilerde olduğu gibi inşaat endüstrisinin de iş yapma biçimini büyük oranda etkilemiştir. BİT uygulamalarından biri olan Yapı Bilgi Modellemesi (YBM) günümüzde gelişmiş ülkelerin Mimarlık, Mühendislik ve İnşaat (AEC) sektöründeki firmalar için vazgeçilmez bir teknoloji olmuştur. Bu sayede yüklenici, işveren, mühendis ve işçiler arasında kurulan güçlü bir iletişim ağı sayesinde iş birliği oluşturulmaktadır. Gelişmiş ülkeler İnşaat projelerine YBM uygulamalarını dahil ederek YBM'nin birçok avantajlarından faydalanmaktadır.

BİT uygulamaları kullanılarak inşaat projeleri daha etkili ve verimli olacaktır. İnşaat endüstrisi için en önemli BİT uygulaması YBM' dir. Özellikle tasarım, planlama ve uygulama için YBM önemli avantajlar sağlamaktadır. BİT uygulamalarının hem gelişmiş hem de gelişmekte olan ülkeler tarafından benimsenmesinde ortak zorluklar vardır. YBM sürecinin uygulanmasındaki ortak engeller arasında mal sahibinin YBM' yi tanımaması ve zorlayıcı kararları bulunmaktadır (Succar, 2010). Fakat güvenlik meselesi gelişmekte olan ülkelere özgüdür. Özellikle gelişmekte olan ülkelerde işletmelerin korsan yazılım kullanmaları birçok güvenlik sorununu beraberinde getirmektedir. Gelişmekte olan ülkelerde kalifiye eleman eksikliği, çalışanların değişime karşı çıkması, kültürel etkiler ve politikalar YBM' in benimsenmesini zorlaştırmaktadır.

2.2 Yapı Bilgi Modellemesi

Yapı Bilgi Modellemesi (YBM), inşaat profesyonellerinin işbirliğine dayanarak bir yapının oluşturulması ve yönetilmesi için bir BİT aracıdır. YBM, proje paydaşları tarafından inşaat projelerindeki birçok hatayı giderebilecek bir araç olarak görülmektedir (Zuppa vd., 2009).

İnşaat Mühendisliği alanındaki profesyoneller tarafından oluşturulmuş birçok YBM tanımı bulunmaktadır. Profesyoneller, farklı görüşlerden doğan farklılıklara rağmen YBM

olarak adlandırılan her modelin hangi bilgileri içermesi gerektiği konusunda hemfikirlerdir. Bu nedenle YBM'nin oluşturulmasını 3B bina modellemesi ve model için gerekli bilgilerin girişi olarak 2 kısma ayırmak mümkündür. Binanın 3B modeli sıfırdan 3B modelleme ya da önce 2B tasarımının gerçekleştirilip ardından çeşitli modelleme uygulamaları ile 3B modelleme olarak çeşitli yollarla tasarlanabilmektedir. Ayrıca YBM sürecinde bilgiler ne kadar ayrıntılı bir şekilde aktarılırsa metraj ve keşif aşamalarındaki işlerin daha fazla ayrıntısı elde edilebilecek ve maliyet hesabındaki hata oranı azaltılacaktır (Zima, 2017).

YBM kullanılarak 3B görünüm ile Metraj Tablosu (MT) hazırlama sürecinin verimliliğini arttırmak amaçlanmaktadır. Geleneksel MT detaylı çizimlere dayanarak hazırlanmaktadır. MT, ihale sürecinde maliyeti tahmin etme ve müteahhitler tarafından sunulan teklif bedellerinde aşırı düşük teklifi veya kırım oranını tespit etmek için kullanılır. İhale sonrası aşamada da maliyet planlaması, nakit akışı projeksiyonu, bütçe tahsisi, ara ödemeler ve kesin hesap mutabakatları için uygulamaları vardır.

Taşınabilir Belge Biçimi (*Portable Document Format PDF*) aracılığıyla yüklenen model üzerinde döndürme, yakınlaştırma-uzaklaştırma işlemleri yapılabilmektedir. Yapının 3B modeli metraj bilgileri ile birlikte PDF dosyasına eklenebilmektedir. Bu sayede proje paydaşlarının inşaat süresi ile ilgili analizleri daha kısa sürede yapabilmeleri ve maliyeti daha az hata ile hesaplayabilmeleri sağlanmıştır. Denetçiler ve öğrenciler arasında yapılan bir anket ile 3D modelin dahil olduğu MT ile standart MT karşılaştırılmıştır. Sonuçlara göre özellikle tahmin için sağlanan görselleştirme ile yüklenicinin Metraj Denetçisinin projeyi daha iyi anlaması bakımından yeni MT modeli daha çok benimsenmiştir. Sonuç olarak yeni MT modeli sayesinde projeler çok daha doğru ve kısa zamanda değerlendirilebilecektir. Maliyet tahminleri daha doğru bir şekilde elde edilecektir. Ayrıca bu formatta belge gönderimi ve alımı oldukça basit güvenli bir işlem olarak tanımlanmaktadır (Nadeem vd., 2015).

YBM inşaat sektöründe yenilik ve üretkenlik için hızlandırıcı bir etkiye sahiptir. Gelişmekte olan ülkelerdeki büyük bilgi eksikliği ve teknolojik yeniliklerin kısıtlı olması gibi mevcut olan çeşitli zorluklar bu ülkelerdeki inşaatların gecikmesine, yapıların düşük kaliteli olmasına, kötü çalışma koşullarına ve kazalara sebep olmaktadır. Belirtilen ülkelerde inşaat endüstrisinden daha yüksek verim elde edildiğinde ülkelerin kalkınma ve büyüme hedeflerinin gerçekleşme olasılığı artmaktadır. Ayrıca yapıların kullanımı boyunca daha düşük ısıtma ve aydınlatma giderine sahip olmasını sağlamaktadır. Bununla

birlikte inşaat sürecinde daha yüksek denetimin uygulanmasını sağlayacağı için inşaatın daha az hata ile yürütülmesini sağlayacak ve yapının deprem başta olmak üzere doğal afetlere karşı daha yüksek dayanıma sahip olmasına yardımcı olur. Belirtilen nedenler sayesinde YBM, gelişmekte olan ülkelerde yoksulluğun ortadan kaldırılmasına katkıda bulunabilecek daha sürdürülebilir bir inşaat sürecine yardımcı olabilecektir.

YBM'nin gelişmekte olan ülkelerde yürütülen inşaat projelerindeki tasarım hatalarını önemli ölçüde azalttığı tespit edilmiş ve inşaat yönetimi sürecinde etkin bir veri yönetimi uygulamasının çakışmaları engellediği ve maliyeti azalttığı gözlemlenmiştir. Yapılan araştırmalarda gelişmekte olan ülkelerin YBM standartlarını ülkedeki standartlara aktarmaya çalıştıkları gözlemlenirken inşaat profesyonelleri tarafından YBM benimseme oranı oldukça düşüktür. Ayrıca bu ülkelerdeki firmalar konvansiyonel proje yönetiminden YBM uygulamasına geçişi riskli bir yatırım olarak görmektedir. Bu yüzden gelenekçi firmalar var olan mevcut durumu değiştirmede isteksizdirler. Gelişmekte olan ülkelerdeki girişimciler, Parametrik Modelleme Teknolojisi (PMT) tabanlı Entegre Proje Teslimi (EPT) ile iş yapmanın becerilerini geliştirmede umut verici bir yol olarak görmektedir. Çin, Hindistan ve Malezya dışında gelişmekte olan ülkeler arasında YBM'yi benimsenme oranının oldukça düşük olduğu gözlenmektedir. Ayrıca gelişmekte olan ülkelerde kullanılan YBM yazılımları, yazılımın eklentileri ve güncelleştirmeleri olmadan özelleştirilmemiş orijinal ilk sürümü olan vanilya yazılımlardır. Gelişmiş ülkelerde YBM uygulamaları yerel ihtiyaçlar ve mevzuatları gereği özelleştirilmektedir.

Gelişmekte olan ülkeler için YBM teknolojisi teknik ve maddi kısıtlardan dolayı emekleme aşamasındadır. Bu ülkelerin büyük bir kısmında YBM üzerinde yürütülen kapsamlı çalışmalar sınırlıdır. Ayrıca bu ülkelerde yapılan çalışmalarda ulaşım (yollar, köprüler vd.) ve altyapı (kanalizasyon, su, doğalgaz vd.) konuları çok sınırlı olarak irdelenmiştir. YBM yazılımlarının ithal olarak gelişmiş ülkelere tedarik edilme fikri benimsenmesinden dolayı yerel ihtiyaçlara uygun bir şekilde projenin tasarlanması zorlaşmaktadır. Devletlerin YBM uygulamalarını teşvik etmesi ve finansal olarak firmalara yardımda bulunması çözüm olarak önerilmektedir. YBM ile ilgili politikalar ve yönetmelikler ülkenin ilgili mevzuatlarına ve yerel ihtiyaçlarına uygun olarak düzenlenmelidir (Bui vd., 2016).

Woo, (2006) öğrencilere verilen YBM kursu sonucunda öğrencilerin daha istekli ve daha düzenli bir şekilde dönem sonu ödevlerini yerine getirdiğini gözlemlemiştir. Ayrıca

öğrenciler kurs sonunda YBM' nin inşaat sektöründe dikkate alınması gereken bir kavram olduğu düşüncesine sahip olmuşlardır. Kurs sonucunda YBM' nin sadece bir BDT programı olmadığı aynı zamanda yapıyı oluşturan birbiriyle ilişkili modelleri sağlayan bir süreç olduğu ortaya çıkmıştır.

Bynum vd., (2013) tarafından gerçekleştirilen bir anket çalışmasında sürdürülebilir inşaat tasarımlarının şirketler için önemli olsa da YBM'nin birincil hedefleri arasında olmadığını ortaya koymuştur. Yapılan ankete göre katılımcılar, YBM'nin başlıca hedeflerinin proje koordinasyonunu sağlama ve görselleştirme olduğunu belirtmişlerdir. YBM'nin daha fazla kullanımı ile yapılar için sürdürülebilirlik konusu da YBM'nin başlıca hedefleri arasına girmesi olasıdır.

YBM sürecinde kullanılan yazılımlar tek başına eklentisiz olarak kullanıldığı durumlarda metraj tahmininin güvenilirliği azalmaktadır. Bilgisayar desteği olmadan yapılan metraj hesabı karmaşıktır ve doğruluk açısından güvenilirliği düşüktür. YBM yazılımları ile gerçekleştirilen metrajlarda metrajı yapılan yapıya ait elemanların özelliklerinin ayrıntılı olarak belirlenmemesi ve farklı durumların göz önüne alınmaması gibi durumlardan dolayı güvenilirlik azalmaktadır. Choi vd., (2015) metraj tahminin güvenilirliğini arttırmak için bir model geliştirmişlerdir. Önerilen metraj hesabı süreci 4 kısıma ayrılmaktadır. Sırasıyla modelleme, veri güvenilirliğini doğrulama, veri kalitesini doğrulama ve metraj hesabının gerçekleşmesi kısımlarından oluşan bu süreç sayesinde hesaplanan metrajın doğruluğu ve kalitesi artmaktadır.

2.3 Önemli YBM Yazılımları

2.3.1 Autodesk Revit

Autodesk firması tarafından geliştirilen bir yazılım olan Revit, “Revise Instantly” (Anında Revize etmek) anlamına gelmektedir. Revit, Yapı Bilgi Modellemesi (YBM) olarak adlandırılan sürecin temelini oluşturan bir yazılımdır. Mimari, yapı, elektrik ve sıhhi tesisat tasarımcıları tarafından kullanılmaktadır. Kullanıcıların bir yapıyı 3B olarak tasarlamalarını ve tasarlanan yapı elemanlarının bilgilerinin bir veritabanında tutulmasını sağlar. Ayrıca Revit, zaman kavramının 3B modele eklenmesi ile oluşan 4B YBM uygulamasıdır. Bu sayede yapının yaşam döngüsü boyunca çeşitli aşamalarının tasarlanmasını ve izlenmesini sağlar. Proje paydaşları arasında iş birliğini kolaylaştıran Autodesk Revit' in ilk sürümü 2000 yılında piyasaya çıkmıştır (Stine, 2012).

2.3.2 ArchiCAD

ArchiCAD yazılımı Graphisoft şirketi tarafından geliştirilip dağıtılan bilgisayar destekli tasarım yazılımıdır. İlk sürümleri sadece Apple tarafından tasarlanan Machintosh işletim sistemlerinin sürümlerinde kullanılıyorken uzun zamandır Microsoft tarafından tasarlanan Windows işletim sistemlerinde de kullanılabilmektedir. YBM sürecinde kullanılan bir yazılım olan ArchiCAD yazılımı diğer yazılımlardan farklı olarak Türkçe dil desteğine sahip nadir yazılımlardandır (ArchiCAD - Vikipedi, 2022).

2.3.3 Diğer yazılımlar

Autodesk Revit ve ArchiCAD dışında, Allplan, Autodesk BIM 360 ve Autodesk Viewer YBM sürecinde kullanılan diğer önemli yazılımlardır. Özellikle Autodesk Viewer, .DWG, .DXF, .RVT ve .IFC gibi dosya formatlarını görüntüleyen ücretsiz online bir yazılımdır. Autodesk Viewer yazılımını ücretsiz kullanabilmek için Autodesk hesabı açılması gerekmektedir. Web ortamında tarayıcılar vasıtasıyla dosya formatları görüntülenebilir. Ülkemizde Sta4CAD ve IdeCAD yazılımları statik-betonarme ve metraj hesaplamaları için kullanılan yazılımlardır.

YBM sürecini mevcut durumda tek bir yazılım kullanarak sürdürmek olanaksızdır. Bu yüzden proje paydaşları proje başında ve proje ilerlerken birden fazla yazılım aracı kullanmaktadır. YBM yazılım araçları giderek inşaat sektörü için vazgeçilmez bir hale gelmektedir. Bu nedenle şirketlerden üniversitelere kadar tüm katılımcıların YBM'yi iyi bir şekilde analiz edip yüksek fayda elde etmek için çalışmalara başlaması gerekmektedir. Gelişmekte olan ülkelerde YBM'nin kullanımının yaygınlaşması ile birlikte iş güvenliğinden kaynak tasarrufuna kadar birçok açıdan inşaat sektörü ileri seviyelere çıkmış olacaktır.

2.4 IFC

Industry Foundation Classes (IFC) inşaat sektöründe bir yapı hakkında kullanıcılar arasında bilgi alışverişinde bulunmak için kullanılan uluslararası bir standarttır. IFC şeması EXPRESS Tanımlama Dili kullanılarak tanımlanmıştır. EXPRESS sayesinde bir alana ait nesneler ve bu nesnelere ait boyut, renk gibi nitelikler tanımlanmaktadır. (BuildingSmart, t.y.).

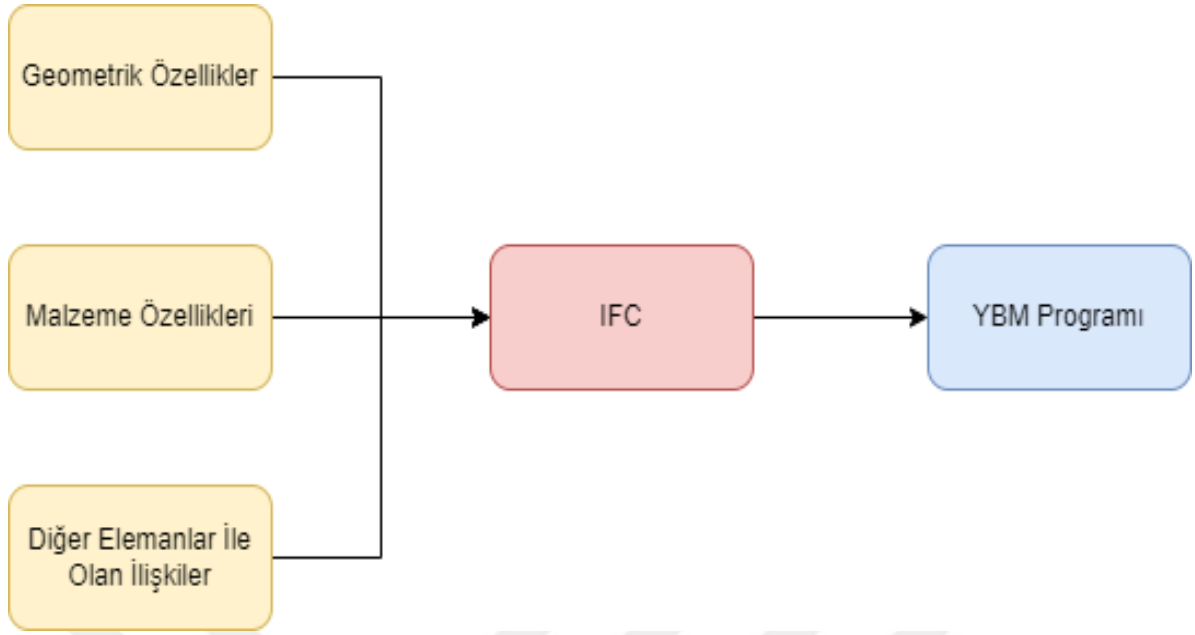
IFC ile bina elemanlarının fiziksel özellikleri, geometrik özellikleri ve elemanın diğer elemanlar ile olan ilişkileri tanımlanabilir. YBM, kullanıcılar için işbirliğini

artırmasına rağmen koordinasyon açısından birçok yeni sorun yaratmaktadır. Proje paydaşlarının her birinin yapı üzerinde farklı modellemeler yapması sonucunda modellerin birleştirilme ihtiyacı ortaya çıkmaktadır. Örneğin bir mühendis yapı elemanlarını modellerken paydaşlardan birisi sıhhi tesisatı modellemektedir. IFC sayesinde bu modeller birleştirilmekte ve karmaşıklık azaltılmaktadır. Proje paydaşları arasında işbirliğini artıran IFC, YBM yazılımları arasında yapıya ait neredeyse tüm bilgilerin değiş tokuş edilmesini sağlamaktadır (Eastman vd., 2011).

Bir yapının inşaatı sırasında Mimarlık, Mühendislik ve İnşaat (AEC) sektöründeki birçok disiplin bir arada bulunmaktadır. Böylece yapı hakkında birçok disiplin ile ilgili bilgileri barındıran IFC, proje paydaşları için birçok gereksiz bilgiyi de barındırabilmektedir. Örneğin Isıtma, Havalandırma ve Soğutma (HVAC) sistemi ile ilgili özelliklerin gerekli olduğu bir durumda diğer elemanlara ait özelliklerin de birarada olduğu durum çizimin karmaşıklığını arttırarak kafa karışıklığı yaratabilmektedir. Kısacası tüm verilerin tek bir dosyada olması istenmeyen bir durumdur. Bu durumu önlemek için IFC, Model Görünüm Tanımları (MVD) adında alt kümelere ayrılmaktadır. MVD, belirli bir kullanım için standartlaşmış veri alışverişi olarak tanımlanmaktadır. Örneğin modelin sadece rengini değiştirmek isteyen kullanıcı için karmaşık model yerine nesnelerin özelliklerinin bulunmadığı basit geometrik model oluşturulur. Böylece IFC kapsamı daraltılır (BuildingSmart, t.y.).

Belsky vd., (2016) geliştirdikleri bir algoritma ile IFC içerisindeki gereksiz bilgileri çok daha hızlı bir şekilde ortadan kaldırıp anlamsal olarak daha zengin bir IFC modeli oluşturmuşlardır. HTML'e gömülebilen, açık kaynaklı ve web geliştirme amacı için kullanılan programlama dili olan PHP ile bir kullanıcı arayüzü geliştirilmiştir. Kullanıcı arabirimi sayesinde oluşturulacak olan zenginleştirilmiş yeni IFC için gerekli olan kurallar kullanıcı tarafından tanımlanmıştır. Ardından tanımlanan kurallar “*IF AND THEN*” yapısını barındıran algoritma ile test edilerek yeni IFC dosyası oluşturulmuştur. Oluşturulan yeni IFC dosyası önceki ile karşılaştırıldığında yapı elemanları hakkında çok daha bilgi düzeyi yüksek ve zengin bir içeriğe sahip olduğu görülmüştür. Günümüzdeki YBM yazılımlarının oluşturduğu IFC dosyalarının zenginleştirilmesi, inşaat projelerinin tüm aşamalarına olumlu bir katkı yapması olasıdır.

IFC ile YBM arasındaki ilişki Şekil 2.1’de gösterilmiştir.



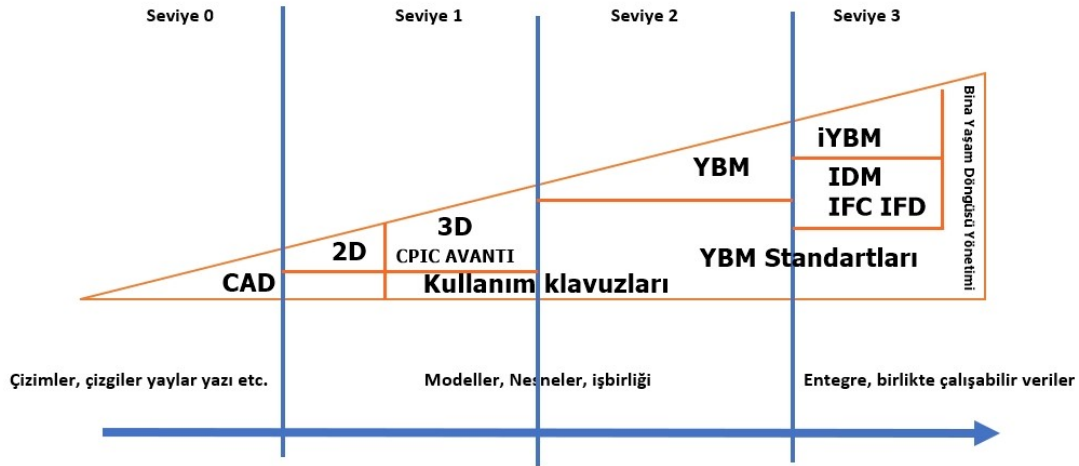
Şekil 2.1: Industry Foundation Classes (IFC) uluslararası veri alışverişi formatının içeriği ve bir YBM yazılımında görüntülenme süreci

2.5 YBM Verimliliği

Projelerde YBM'nin fayda sağlama potansiyelini tespit etmek için ortak bir YBM tanımı kabul edilmelidir. YBM'nin “iş birliği için açık standartlar üzerine inşa edilmiş paylaşılan bir dijital temsil” olarak tanımlanması, proje paydaşları arasında bir ayrım yapmadan genele hitap eden bir tanım olduğu için uygun görülmüştür (Barlish ve Sullivan, 2012).

YBM kapasitesinin mükemmelliğinin derecesi olarak tanımlanan “YBM Olgunluk Seviyesi” (YOS), bir organizasyonun YBM'yi tanımlaması ve anlamasını etkileyen önemli bir faktördür. Kuruluşların YBM iş vakalarını karşılaştırırken YOS dikkate alınmalıdır (Barlish ve Sullivan, 2012). YOS tanımlaması Şekil 2.2'de gösterilmiştir.

Bu çalışmada geliştirilen yazılım, metraj hesaplamaları, 3B görselleştirme ve dxf formatında export seçeneği sunma gibi özellikleri incelendiğinde YOS seviye 1 ve seviye 2 arasında bir konumda olduğu görülmektedir.



Şekil 2.2: YBM Olgunluk Seviyeleri (Building Information Modelling (BIM) Working Party Strategy Paper, 2011)

Barlish ve Sullivan (2012), YBM kullanımının yararlarını belirlemek için bir çerçeve oluşturmayı hedeflemişlerdir. Literatür taraması sonucunda en çok bahsedilen YBM yararlarını tespit ederek ölçekler oluşturmuşlardır. Bu ölçekler içerisinde iş programı, maliyet ve sipariş değişiklikleri gibi öznel olmayan metrikler belirlenmiştir. Seçilen ölçekleri YBM'nin kullanıldığı projeler ile YBM'nin kullanılmadığı projelerde üç farklı vaka çalışmasında test etmişlerdir. Test sonucunda elde edilen verileri değerlendirmişlerdir. Sonuç olarak YBM ile gerçekleştirilen projelerin inşaat maliyetlerinde en az %5 tasarruf edildiği belirlenmiştir. Çerçeve oluştururken BT yararlarını ölçme sürecinden yararlanılmıştır.

İki boyutlu çizim için oluşturulan BDT programları inşaat sektörü için yetersiz kalmaktadır. İnşaat profesyonelleri arasında iletişim kopukluğunu ve koordinasyonsuzluğu gideremeyen BDT programları nedeniyle inşaat sektörü yeni bir otomasyon sistemi arayışına girmiştir. Yapı Bilgi Modellemesi (YBM) ile özellikle iletişim kaynaklı sorunların çözülmesi hedeflenmiştir.

Metraj hesabı günümüzde birçok inşaat şirketi tarafından geleneksel hesaplama yöntemleri ile hesaplanmaktadır. 2D çizimden alınan yapının boyutu ve elemanların geometrileri ile ilgili verilerin MS Excel programına aktararak yapılan hesaplamalar hatalı hesaplama yapma olasılığını artırmaktadır.

Yapıların yaklaşık maliyetleri metraj değerleri ile birim maliyetlerin çarpılmasıyla hesaplanır. Geleneksel yöntemler kullanılarak elde edilen yaklaşık maliyet için hata payı oldukça yüksektir. Bu tarz yöntemlerde genellikle benzer projelerle karşılaştırma ya da belirli bir katsayı ile metraj değerlerinin çarpılması şeklinde kabullere göre yaklaşık maliyet hesaplanmaktadır. İnşaat sürecinde görülebilecek hatalı miktardaki malzeme siparişi, yetersiz nakit akışı gibi sorunların önüne geçmek için yaklaşık maliyet en az hata ile hesaplanmalı ve gerektiğinde güncellenmelidir. Yapılan çalışmalarda Autodesk Revit gibi YBM programları ile gerçekleştirilen yaklaşık maliyet hesaplarının tahminlere dayalı modellere göre daha doğru sonuçlar verdiği gözlemlenmiştir. Özellikle yapının inşa aşamasında gerçekleşen değişiklikler sonucu metraj ve yaklaşık maliyet değerlerinin YBM uygulamaları kullanılarak güncellenmesi YBM uygulamalarının maliyeti azaltmadaki etkisini ve sağladığı verimlilik artışını gözler önüne sermektedir. Sonuç olarak doğru bir inşaat bütçesi oluşturmak için YBM yazılımları kullanılmalı ve yazılımlar güncel tutulmalıdır. YBM süreci gelişim aşamasındadır ve ülkemizdeki inşaat firmaları arasında kullanımı artırılmalıdır (Karagöz, 2019).

Bryde vd. (2013) maliyet yönetiminde YBM sürecinin ve araçlarının kullanımından kaynaklanan olumsuz etkilerin YBM sürecinin ve araçlarının kullanımından kaynaklanan olumlu etkilere göre daha az olduğu belirlenmiştir. Ayrıca YBM, inşaat projelerinin yönetimine yardımcı olan bir araç olarak görülmektedir. YBM sürecinde kullanılan yazılımların maliyetleri ve yazılımlar arasındaki birlikte çalışabilirlik sorunları dikkate alınması gerekli olan önemli konulardır.

Projenin başarılı olma olasılığını arttırmada maliyet kontrolü önemli bir adımdır. Mühendislik projelerinde ekonomik fayda elde edebilmek için projelerin maliyetini azaltmak gerekmektedir (Du, 2021).

İnşaat projelerinin gerçek fiziksel koşullarını dikkate almadan üretim verimliliğinin kesintisiz olarak tahmin edilmesi projelerde maliyet artışlarına ve gecikmelere sebep olmaktadır. 4D YBM inşaat verimliliğini artırmak için oldukça kullanışlıdır. 4D YBM projedeki elemanlar arasındaki çakışmaların tespitinde ve inşaatın iş sıralamalarını simüle etmek için kullanılır. Khataei vd. (2020) 4D BIM'den yararlanarak inşaat sırasında projedeki elemanların yakınlıklarını hesaplayarak verimlilik değerlerini revize etmeyi amaçlamıştır.

Khataei vd. YBM ile gerçekleştirdikleri tasarımda yapısal ve mimari elemanlar ile bu elemanların özniteliklerini belirlemiştir. Dynamo görsel programlama dili kullanılarak elemanlara özgün kimlikler atayıp, Dynamo kullanılarak ilgili kimlik ve veri ile elemanları tablo şeklinde MS Excel programına aktarmıştır. Aktarılan verilerle Excel’de saat bazlı bir program oluşturup verimlilik değerlerini atayıp, YBM elemanları ile tablo eşleştirmiş ve Python programlama dili kullanılarak yakınlık kontrolü yapmıştır. Yakınlığa göre verimlilik düşüşü tespit edilir ve Dynamo ile yeni bir verimlilik listesi hesaplanıp kaydedilir. Yakınlık tespiti öncesinde kullanıcı tarafından elemanlar arasında bir eşik mesafesi ve eşiğin aşılması durumunda verimlilik azaltma oranı belirlenmektedir. Her bir çalışma saati içerisindeki elemanların yakınlık miktarları Python ile yazılan bir kod parçacığı ile tespit edilir. Yakınlık miktarlarına göre kullanıcı tarafından girilen verimlilik azaltma oranına göre toplam adam.saat verileri güncellenir. Güncellenen değerler Dynamo aracılığıyla daha önce oluşturulmuş olan Excel tablosuna aktarılır ve iş programı güncellenir. Elemanlar arasındaki yakınlık miktarları dikkate alınarak verimlilik değerlerini revize eden ve iş programını güncelleyen bir model geliştirilmiştir. Gerçek yaşam koşullarına göre güncellenen iş programı sayesinde daha doğru başlangıç tahminleri yapılabilmektedir. Bu sayede projedeki maliyet artışları ile gecikmelerin önüne geçilmesi amaçlanmaktadır. Fakat yakınlık tespiti sadece mevcut olan yapı elemanları ile sınırlı kalmıştır. Kaynak ve iş makinelerini de kapsayacak bir yakınlık tespiti ile daha gerçekçi bir iş programı oluşturulabilir (Khataei vd., 2020).

Azhar vd. (2012a) YBM’nin Mimarlık, Mühendislik ve İnşaat (AEC) Endüstrisini yeniden şekillendiren hem bir teknoloji hem de bir süreç olduğunu ifade etmektedir. Projenin planlama, tasarım, inşaat ve işletme aşamalarındaki gerekli bilgileri bulundurması ve bu bilgilerin 3B model şekline dönüştürülmesi YBM’nin teknoloji kısmını oluşturmaktadır. Yapı sahibi, tasarımcı, mühendis ve yüklenicilerden oluşan proje paydaşlarını bir araya getirerek geleneksel yöntemle göre daha doğru ve verimli bir şekilde iş birliği yapmalarını sağlaması YBM’nin süreç kısmını oluşturmaktadır. YBM, bir projenin inşaat öncesi, inşaat ve inşaat sonrası aşamalarında önemli rol oynamaktadır. Azhar vd. (2012a) bir vaka çalışmasında bir projenin inşaatı öncesinde binanın 2B yapı, mekanik, elektrik ve sıhhi tesisat çizimlerini tek bir 3B YBM modeli ile bütünleştirip çakışma analizi ile çakışmaları tespit etmişlerdir. Elde edilen çakışmalar sonucu yaklaşık 259 bin dolar tasarruf elde edilmiştir. YBM proje paydaşları için birçok avantaja sahip olmasına rağmen uygulanmasında bir takım riskler bulunmaktadır. YBM uygulamaları

arasında geiř yaparken modellerin hasar grmesi ve lisans problemleri teknolojik riskleri oluřtururken szleřme sorunları ile veri giriřini kimin yapacaėı ve kimin hatalardan sorumlu olacaėı gibi belirsizlikler sre ile ilgili riskleri oluřturmaktadır (Azhar vd., 2012a).

Azhar, (2011) inřaat ncesi řematik tasarımı ařamasında YBM kullanılarak mal sahibinin istekleri doėrultusunda 3 farklı tasarım ve bu tasarımlara uygun olarak 3 farklı maliyet senaryosu hazırlanmıřtır. Tasarımlar arasından mal sahibinin gereksinimlerine uygun olan biri seilerek yaklaşık 2 milyon dolar maliyetten tasarruf edilmiřtir. YBM, inřaat ncesi proje tasarımında kullanılarak hızlı ve bilinli bir karar verilmesini saėlamıřtır.

Azhar vd. (2012b) YBM'yi kullanarak bir binanın iř gvenliėinin ynetimini gerekleřtirmiřlerdir. řantiye ierisinde gerekleřme olasılıėı bulunan tehlikelere karřı alıřanları uyarmak iin 4 Boyutlu animasyonlar yapılmıřtır. Denetiler, YBM'yi bir eėitim aracı olarak kullanmıřlardır.

3D model tabanlı iřbirliėinin etkilerini gzlemlemek iin yapılan vaka alıřmaları sonucu elde edilen bulgular, geometrik model kontrolnn tasarım ile inřaatın yapımı sırasındaki iř blm ile iliřkili olduėunu gstermektedir (Ku vd., 2008). Model tabanlı iřbirliėinin verimli bir řekilde gerekleřmesi iin gereken iki faktr řunlardır:

- Yapının belirli ařamalarında proje paydařlarını dahil etmek iin teknik yeterliliėinin bulunması.
- Tasarım ve inřaat kısımlarındaki sorumlulukların uygun řekilde tanımlanması.

Model tabanlı geometri kontrolnn potansiyelini artırmak iin mimari uygulamaya ve tasarım liderliėini kazanma ve srdrmeye izin veren yaklařımlar gereklidir (Ku vd., 2008).

YBM sreci, projelerde sadece tasarım ve izim ařamalarını deėil aynı zamanda yařam dngs srecinde bulunan yapım ve bakım-onarım ařamalarını da kapsamaktadır. rneėin havalimanları gibi yapılarda srdrebilirlik nemli olduėu iin bakım-onarım ařamalarında YBM sreci nemli rol oynamaktadır (İnceoėlu ve İnan, 2020).

2.6 YBM Sürecinde Metraj ve Maliyet Tahmini

Metraj hesabında kullanılmak için yapı elemanının modelinden elde edilen veriler, hesabın doğruluğu açısından oldukça önemlidir ve maliyeti de doğrudan etkilemektedir. YBM ile hesaplanan metrajın geleneksel yöntemler ile hesaplanan metraja göre daha hızlı hesaplanmasına rağmen geometrik modelin tanımındaki farklılıklar, YBM ile elde edilen metraj sonuçlarının doğrudan kullanımını olumsuz yönde etkilemektedir (Yang vd., 2019).

Kim vd. (2019) YBM sürecinde hesaplanan metraj değerlerindeki tutarsızlığın malzemelerin türüne, kullanım amacına ve modelleme metoduna göre değiştiğini tespit etmişlerdir. Kore’de farklı alanlara sahip olan 4 konut projesi, Bileşik Modellenen Nesne (BMN) ve Ayrık Modellenen Nesne (AMN) metodları kullanılarak herbiri 2 farklı şekilde modellenmiştir. Yalıtım malzemeleri, karo ve boya gibi sadece iç mekân elemanları kullanılarak yapıların her modelleme metoduna özgü olarak metraj değerleri tespit edilmiştir.

Ardından Denklem 2.1 kullanılarak Metraj Tutarsızlığı (MeT) elde edilmiştir. Elde edilen sonuçlara göre BMN modeline ait MeT değeri, AMN modeline ait MeT değerinden %6-9 daha fazladır. Bu yüzden metraj değerlerinin düşük hesaplandığından şüphe duyuluyorsa nesneler ayrık bir biçimde modellenmelidir.

$$MeT = \frac{MBMN - MAMN}{MAMN} \quad (2.1)$$

Liu vd. (2022) YBM sürecinde kullanılan programlar ile gerçekleştirilen metraj hesabındaki hatayı en aza indirmek için bilgi modeli tabanlı bir çerçeve sunmuşlardır. YBM sürecinde modellenen elemanların sadece geometrik özellikleri bilinmektedir. Sunulan bu çerçeve ile modeller için anlamsal bir temsil oluşturularak beton sınıfı gibi özellikler modele eklenmiştir. Anlamsal ve geometrik özellikler kullanılarak hesaplanan metraj değerlerinin YBM sürecinde kullanılan Revit ve Cubicost programları ile hesaplanan metraj değerlerinden daha az hatalı olduğu tespit edilmiştir. Sonuç olarak daha doğru bir metraj elde edebilmek için ilgili elemanların geometrik modeli dışında anlamsal bir temsiline de oluşturulması gerekmektedir.

Ismail vd. (2016) yapmış oldukları bir çalışmada YBM sürecinin sunduğu yararları belirlemek için bir çerçeve geliştirmişlerdir. Geliştirilen çerçeve sonucunda YBM kullanımı ile maliyet tahminlerinin güvenilirliğinin artışı arasında bir ilişki olduğunu

belirlemişlerdir. Böylece, inşaat projelerinde maliyet tahmini için YBM'den faydalanılarak metrajcılarının ve araştırmacıların kavrama ve anlayış becerileri geliştirilebilir.

Thurairajah ve Goucher, (2013) tarafından yapılan bir araştırmaya göre inşaat sektöründe YBM kullanımı her açıdan sektörü etkileme potansiyeline sahiptir. YBM kullanımı özellikle maliyet tahmini sırasında büyük avantajlar sunmaktadır. Gelecekte YBM'yi benimseyecek olan şirketler, YBM'nin avantajlarından faydalanarak becerilerini geliştirecekleri tahmin edilmektedir.

Birleşik elemanlar içeren bir yapının her odası için gerekli olan duvar ve zemin materyallerinin metrajı, YBM sürecinde kullanılan en önemli yazılımlardan biri olan Autodesk Revit ile eklentisiz olarak ve Khosakitchalert vd. (2020) tarafından Dynamo uzantısı kullanılarak geliştirilen prototip sistem ile ayrı ayrı hesaplanmıştır. Hesaplanan metraj değerleri karşılaştırıldığında Dynamo uzantısı ile geliştirilen sistemin Revit'e eklenmesi ile daha yüksek doğrulukta metraj hesaplanmıştır. Ayrıca geliştirilen prototip sistemi ile yapı üzerinde örtüşen elemanlar daha kısa sürede tespit edilerek zamandan da tasarruf edilmiştir. Böylece YBM sürecinde kullanılan yazılımların tek başına yeterli olmadığı sonradan geliştirilen algoritmalar ya da sistemler ile YBM sürecinin güçlendirilmesinin gerekli olduğu ortaya çıkmıştır.

YBM ile bir binanın 3B olarak modellenmesi ile 3B YBM elde edilir. Projenin tamamlanması için gerekli olan sürenin tespit edilmesi ile birlikte zaman kavramının YBM sürecine eklenmesi sonucu 4B YBM elde edilir.

Planlama dışında 4B YBM güvenlik, atık yönetimi gibi alanlarda da planlayıcılara yardımcı olmaktadır. Jupp, (2017) tarafından yapılan bir araştırma çevresel planlama ve yönetim ile ilgili çalışmaların 4B modelleme yetenekleriyle birleştirilip geliştirilebileceğini ortaya koymuştur.

YBM modeli içerisindeki nesneleri maliyet verileri ile ilişkilendirerek YBM sürecine 5. Bir boyut katılmış olur ve böylece 5B YBM ortaya çıkmaktadır (Boon ve Prigg, 2012). İnşaat ile ilgili detayların ve çizimlerin görselleştirilmesi ve detaylandırılması ile birlikte daha ileri seviyede risk tespiti yeteneğine sahip olan 5B YBM geleneksel yöntemler ile yapılan tahmin ve maliyet analizlerine karşı büyük avantajlara sahiptir. YBM sürecinde kullanılan yazılımların maliyeti, YBM yazılımlarını kodlamak için geçerli olan

bir standardın olmaması ve YBM ile ilgili önyargılar, 5B YBM'nin kullanımının yaygınlaştırılmasını engellemektedir (Stanley ve Thurnell, 2014).

İnşaat sektöründe YBM' nin kullanılması daha doğru metraj hesabı, inşaat sırasında meydana gelebilecek olan değişiklik taleplerinin azalması, proje paydaşları arasında bilgi taleplerinin azalması, paydaşlar arasında güçlü iş birliği ve zaman tasarrufu gibi faydalar sağlamaktadır (Franco vd., 2015).

2.7 YBM ve Oyun Motorları

Yapı Bilgi Modellemesi (YBM), 4D modellemeden yapısal analize kadar birçok alanda kullanılmaktadır. YBM programları arasında birlikte çalışabilirlik açısından sorunlar meydana gelmektedir. Standartlaşma YBM programlarının birlikte çalışabilmesi için olası bir çözümdür. (Endüstri Temel Sınıfları) ETS IFC ana standartlaşma yöntemlerinden biridir.

Zhang vd. (2019) IFC'de mimarlık, mühendislik, yapım (AEC) nesnelerinin geometrik temsillerinin analizi için Oyun Simülasyonunda Kartezyen Nokta Görselleştirilmesini gerçekleştirmiştir. 14 kişiden oluşan bir katılımcı kitlesine yazılı açıklama materyali ile oyun simülasyonu verilerek yapılan deneyde oyun simülasyonu ile yapılan görselleştirmeyi anlamamanın yazılı açıklamaya göre oldukça kolay ve daha az vakit aldığı tespit edilmiştir. Sonuç olarak Unity3D, Unreal Engine gibi oyun motorları ile Kartezyen noktaları görselleştirmesi yapılabilmektedir. Deneyde birinci şahıs bakış açısı kullanılmıştır. Diğer bakış açıları kullanılarak ve deneye katılan kişi sayısı artırılarak sonuçlar geliştirilebilir.

YBM sürecindeki yazılımları kullanan proje paydaşlarının özellikle veri alışverişi sırasında yerel mevzuatlara uygun olacak biçimde standartlar geliştirmelidir. Öte yandan bir inşaat projesi boyunca meydana gelen değişiklikleri görselleştirerek sunan YBM, proje paydaşlarına büyük bir avantaj sunmaktadır (Sabol, 2008).

YBM tabanlı çalışmalar özellikle Avrupa'da büyük oranda artmaktadır. Günümüzde YBM ile yapılan projelerde Yaşam Döngü Maliyeti (YDM) hesabı içeren proje sayısı oldukça azdır. Genelde yapının başlangıç maliyeti dikkate alınıp işletme maliyeti vs. dikkate alınmamaktadır. YDM, YBM ve oyun motorlarının entegre olarak bir arada bulunduğu çalışmalarda çok azdır. YBM, farklı disiplin alanlarının ortak bir çalışma üzerinde gözlem yapmalarını ve etkileşim kurmalarını sağlar. Omaran vd. (2019) Unity3D

oyun motoru kullanılarak YDM görselleştirilmesi yapmıştır. Tasarlanan yapı Revit, SketchUp vb. yazılımlarla uyumlu FBX dosya formatına dönüştürülmüştür. Ardından materyal ve kaplamaları Unity3D oyun motoruna uyumlu formata dönüştürmek için FBX dosyası 3DS Max programı ile açılıp materyaller dönüştürülmüş ve tekrar FBX formatında çıktı alınmıştır. FBX formatındaki dosya Unity3D'ye aktarılıp oyun motorunda ışık, kamera, 3.şahıs kontrolleri vs. eklenmiştir. Kullanıcı oyun motoru vasıtasıyla yapıyı 3D olarak görebilir ve etrafında gezinebilir duruma getirilmiştir. Aynı anda kaplamaları vs. değiştirilebilir ve yapılan değişikliklere eş zamanlı olarak YDM hesabı da baştan yapıp sonuç kullanıcıya sunulabilmektedir. Kodlar C# ve JavaScript ile yazılmıştır. Oyun motoru kullanılarak yapılan bina simülasyonu YDM ile birleştirilmiştir. Yapılan ankette tasarlanan uygulamayı deneyimleyen kullanıcıların yapı ile ilgili kararını büyük oranda etkilediği ve kararların daha bilinçli bir şekilde alındığını göstermektedir.

Oyun motorları ile birlikte YBM yazılımların birlikte çalışması sonucunda YBM sürecinin proje paydaşları için çok daha verimli bir hale geldiği görülmektedir. Oyun motorları aslında kodlanmış birer fizik motorları olduğu için yapının görselleştirilmesi dışında da birçok konuda YBM sürecinde kullanılabilir. Örneğin YBM tabanlı tasarımlardaki çakışma analizi oyun motorları kullanılarak gerçekleştirilebilir. Bu çalışmada açık kaynaklı bir oyun motoru olan Ursina oyun motoru kullanılmıştır.

2.8 Python Programlama Dili

Python programlama dili, üst düzey ve yorumlanmış genel amaçlı bir dildir. Python, C dilinde yazılmıştır. Python programlama dilindeki girinti kullanımı sayesinde kod okunabilirliği önemli ölçüde arttırılmaktadır. Nesne yönelimli programlama ve fonksiyonel programlama paradigmalarını destekleyen python, otomatik çöp toplama (garbage collection) özelliğine sahiptir. Çöp toplama özelliği sayesinde kullanılmayan nesneyi bellekten siler ve bu sayede performansı arttırır. İlk sürümü Guido van Rossum tarafından çıkarılmıştır ve günümüzde Python 3.x sürümü en son sürümüdür. Geniş bir geliştirici topluluğuna sahip olduğu için sürekli olarak bir gelişim içerisinde (Van Rossum, 2007).

Python, bilimsel topluluğun hesaplama ihtiyaçlarını karşılamak için özel olarak tasarlanmamıştır ancak bilim insanlarının ve mühendislerin ilgisini hızla çekmiştir. Python giderek gelişerek artan ihtiyacı karşılamaya yönelmiştir. Karmaşık sayıların eklenmesi gibi

çeşitli değişiklikler olsa da birçok değişiklik dizi manipölasyonu için daha özlü ve okunması daha kolay bir sözdizimi sağlamaya odaklanmıştır (Millman vd., 2011).

İnşaat Mühendisliği alanında özellikle şantiye yönetimi ve organizasyonu sürecinde hesap tablosu uygulamaları kullanılarak problemlere çözüm aranmaktadır. Fakat hesap tablosu uygulamalarında hücre sayısının artması sonucunda program aşırı yavaşlamaktadır. Ayrıca program içinde inşaat mühendisliği hesaplamalarına yönelik sınırlı sayıda fonksiyon bulunması neticesinde uygulamaların ara hesaplamaları farklı hücrelerde yapılmakta, bu durum ise dosya boyutunu arttırmakta ve programı yavaşlatmaktadır. Bu tez çalışmasında tamamen ücretsiz olan Python programlama dili kullanılarak piyasadaki programların yerine getirdiği işlevler gerçekleştirilecektir. Bu sayede ticari programların aksine kişiselleştirilebilir ve ölçeklenebilir bir uygulama yapılmış olacaktır. Python programlama dili veri bilimindeki en popüler dildir. Scipy, Numpy, Pandas ve Matplotlib kullanılarak kanalizasyon sistemi için bir bölgenin nüfus tahmini, yapısal elemanlar için sonlu elemanlar yöntemi, Yapım yönetimi gibi problemlerin çözümünde kullanılabilir. Mühendislikte kaynak ve zaman kullanımı oldukça önemlidir. Python dili kullanılarak problemi kısa süre içerisinde daha az maliyetle çözmek mümkündür (Quraishi ve Dhapekar, 2021).

İnşaat Mühendisliği problemlerinin çözümünde uygulanan hesaplama süreci karmaşıktır ve çözülmesi zaman almaktadır. Özellikle şantiye gibi kaotik bir ortamda hesap tablosu uygulamaları olsa bile hesaplamaları yapmak oldukça zordur. Python kullanarak bu hesaplamaları daha kısa sürede ve doğru bir şekilde yapabilir. Amaç, kodlamayı basite indirgemek ve sonuca daha hızlı varmaktır. Python paket dizilimi (PyPi) ile diğer diller ile etkileşime girmesini sağlayan modüller yüklenebilmektedir. Ayrıca öğrenmesi kolay ve desteği mevcuttur.

İnşaat Mühendisliğindeki karmaşık problemleri çözebilecek yeterli sayıda kütüphane ve fonksiyon içermektedir. Birçok modül (scipy, tkinter, matplotlib vs.) kullanılarak hızlı ve daha az zaman harcayarak inşaat mühendisliği alanında problemler çözülebilmektedir (Turda ve Gobesz, 2005).

Python, bilimsel topluluk için birçok yararlı kütüphanelere sahiptir. Bunlardan bazılarına NumPy, SciPy, Pandas ve Matplotlib örnek olarak verilebilir. NumPy, python dilinin en önemli kütüphanesidir. Diğer kütüphaneler NumPy kütüphanesi kullanılarak

oluşturulmuştur. Çok boyutlu diziler oluşturmaya yardımcı olur. SciPy, bir diğer önemli kütüphane olan SciPy ile matematiksel işlemlerin gerçekleştirilebilmesi için çok sayıda fonksiyon bulunmaktadır. Pandas, veri analizi ve veri ön işlemeyi kolaylaştırmak için kullanılan bir kütüphanedir. MS Excel hesap tablolarında büyük verilerin işlenmesini sağlar. Matplotlib ise 2B grafikler çizmek için kullanılan bir kütüphanedir. Birçok tarzda yüksek kaliteli grafikler çizilebilir (Sarvade ve Pore, 2019).

Python, programlamaya giriş için oldukça uygun bir programlama dilidir. Bogdanchikov vd., (2013) tarafından gerçekleştirilen bir deneye göre Python ile işlenen derslerin sonucunda gerçekleştirilen ara sınavın not ortalaması java programlama dili ile işlenen derslerin sonucunda gerçekleştirilen ara sınavın not ortalamasından %16 daha fazla olduğunu tespit etmiştir. Programlamaya duyulan arzu ve istek deney sonucunu etkileyen önemli parametrelerdir.

Python programlama dilinin söz dizimi yapısı diğer dillerle karşılaştırıldığında anlaşılması daha basittir. Ayrıca python geliştirme topluluğu dünyadaki en büyük açık kaynaklı yazılım geliştirme topluluğundan birisidir. Böylece devamlı kendini yenileyen ve güncelleyen bir dil olmuştur. Yukarıda açıklanan avantajları ve gelecekte programlama dilleri arasında kullanımı ve geliştirilmesi açısından lider olma potansiyelinden dolayı geliştirilen yazılımın kodlanması için seçilen programlama dili python olmuştur. Ayrıca mühendislik problemlerinde çözüme daha hızlı ulaşmak birçok açıdan tasarruf sağlamaktadır. Python tabanlı kodlama ile yazılım geliştirilerek zamandan tasarruf edilmesi ve istenilen hedeflere daha hızlı ulaşmak amaçlanmıştır.

Bu tez kapsamında geliştirilen programın oluşturulmasında tkinter, math, ursina, random gibi kütüphaneler kullanılmıştır. Ursina dışındaki bahsedilen diğer kütüphaneler python programlama dilinin cihaza kurulması ile birlikte gömülü olarak gelen kütüphanelerdir Tkinter kütüphanesi içerisindeki Tk(), Canvas(), Frame() ve Button(), Toplevel() gibi sınıflar kullanılarak grafiksel kullanıcı arayüzü oluşturulmuştur. Tk() sınıfı, kullanıcı arayüzündeki tüm nesneleri barındıran kök (ana) pencerenin oluşturulmasını sağlamaktadır. Frame() sınıfı, arayüzde gruplanan nesnelerin bir arada olmasını sağlayan adeta sepet görevi gören bir sınıftır. Örneğin butonlar için bir “Frame” nesnesi oluşturularak arayüzdeki butonların gruplanması sağlanmıştır. Canvas() sınıfı ise çizim işlemlerin gerçekleşmesi için oluşturulan bir tuvaldir. Canvas() sınıfı içerisindeki create_rectangle, create_line, create_text metotları ile kullanıcı tarafından alınan veriler

doğrultusunda çizimler gerçekleştirilmektedir. Örneğin `create_line()` sınıfı içerisine 4 parametre verilerek (x1, y1, x2, y2) oluşturulacak olan çizginin başlangıç ve bitiş koordinat değerleri belirlenir ve çizgi `Canvas()` sınıfı ile oluşturulan nesne (tuval) üzerine çizilir. Aks çizimlerinde `create_line` metodundan yararlanılmıştır. `TopLevel()` sınıfının kullanımı ile açılır pencereler oluşturulmuştur. Örneğin kolon butonuna basarak kolonun en, boy gibi özelliklerinin girildiği pencere bir açılır penceredir. Açılır pencere içerisinde `Label()` ve `Entry()` sınıfları kullanılarak kullanıcıdan verilerin girilmesini sağlayan metin kutucukları oluşturulmuştur.

`Math` kütüphanesi, mühendislik hesaplamalarındaki matematiksel işlemler için kullanılmıştır. Örneğin sayıları yukarı yuvarlamak için `ceil()`, aşağı yuvarlamak için `floor()`, verilen bir sayının üssünü almak için `pow()` fonksiyonlarından yararlanılmıştır. Geliştirilen yazılımın donatı metrajında etriyelerin açıklık ve sarılma bölgelerindeki sayısı belirlenirken `ceil()` ve `floor()` metotlarından yararlanılmıştır. Ayrıca mantıksal operatörlerden de (`and`, `or`, `!=`, `==`, `<=`, `>=`, `<`, `>`) yararlanılmıştır. Örneğin döşemenin kaç doğrultuda çalıştığını tespit ederken (uzun kenar/kısa kenar ≤ 2) mantıksal sınaama operatöründen yararlanılmıştır. Mantıksal operatörler daha çok `if-elseif-else` bloklarında kullanılmaktadır.

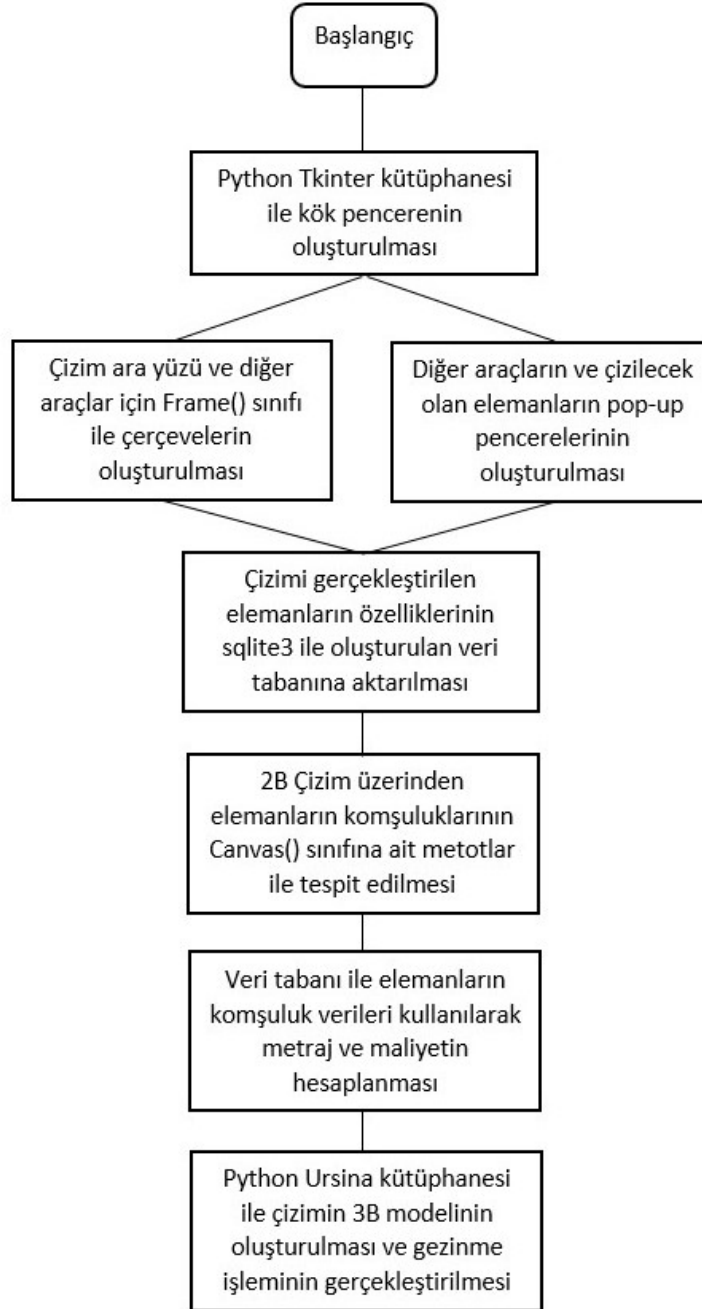
`Random()` fonksiyonu ile istenilen aralıkta rastgele sayı üretilmektedir. Örneğin “`random.randint(3, 9)`” ifadesiyle `random` kütüphanesi çağırılarak içerisindeki `randint` metodu çalıştırılmıştır. Ardından 3 ile 9 arasında rastgele bir sayı döndürmesi sağlanmıştır. `Random()` sınıfı, döşeme donatı planı hazırlanırken herbir döşeme üzerinde düz ve pilye donatılarının yerleştirilmesi için kullanılmıştır.

`Ursina`, Python dili ile geliştirilmiş ücretsiz ve açık kaynaklı oyun motoru amacıyla geliştirilmiş bir kütüphanedir. Yapının görselleştirilmesi ve yapı içerisinde kullanıcının gezinme işlemleri `Ursina` kütüphanesi içerisindeki fonksiyonlar ve metotlar kullanılarak sağlanmaktadır. `Ursina` içerisindeki `Ursina()` sınıfının kullanımı `Tk()` ile aynıdır. Kök pencere ya da kök uygulama oluşturmaktadır. `Entity()` sınıfı ile 3B varlıklar oluşturulmuştur. `EditorCamera()` sınıfı ile kullanıcının yapı içerisinde gezinmesi için bir kamera nesnesi oluşturulmuştur.

Yukarıda bahsedilen metot ve sınıflar dışında geliştirici tarafından yazılan fonksiyonlar da kullanılmıştır.

3. YAZILIMIN GELİŞTİRİLMESİ

Python programlama dili içerisinde bulunan tkinter Kütüphanesi, Python tabanlı Ursina 3B oyun motoru ve SQLite3 veri tabanı kullanılarak Yapı Bilgi Modellemesi tabanlı inşaat planlaması ve 3B görselleştirme işlemlerini gerçekleştiren bir programın geliştirilmesi amaçlanmaktadır. Geliştirme sürecinin akış şeması Şekil 3.1’ de sunulmuştur.



Şekil 3.1: Geliştirilen Yazılımın Akış Şeması

Kullanıcının yazılım ile daha rahat etkileşime geçebilmesini sağlamak ve modern bir uygulama geliştirmek için konsol uygulaması yerine interaktif kullanıcı arayüzüne (GUI) sahip bir yazılım geliştirilmektedir. Python tkinter uygulamalarında ana pencere kök pencere olarak adlandırılır. Diğer açılan tüm pencereler kök pencereye bağlı olduğu için bu isimle nitelendirilmektedir. Kök pencerenin oluşturulması GUI sürecinin ilk adımını oluşturmaktadır. Kök pencerenin ardından kullanıcının yapı elemanlarını çizibilmesi için çizim arayüzü geliştirilmektedir. Çizim arayüzü ile birlikte elemanların çizilmeden önce en, boy gibi özelliklerin girdi kutularını barındıran pop-up pencereleri oluşturulmaktadır. Çizimi yapılan elemanların özellikleri ve koordinat bilgileri sqllite3 ile oluşturulan veritabanına aktarılmaktadır. Verilerin kolay bir şekilde saklanması, çağırılması ve düzenlenebilmesi için veritabanı kullanmak gerekmektedir. Python programlama dili içerisindeki list(), set() gibi fonksiyonlar verilerin saklanması için kullanılabilir. Fakat veriler listede veya bir sözlükte depolanması durumunda devamlı bellekte yer kaplayacaktır. Veritabanı kullanılmasıyla depolanan veriler sadece gerekli olduğu anda çağırılarak gereksiz yere belleği işgal etmesinin önün geçilmiştir. Kısacası veri trafiğinin azaltılması sağlanacaktır. Ayrıca inşaat projesindeki yapı elemanlarının ve özelliklerinin sayısı göz önüne alındığında veritabanı kullanılmadığı zaman bellek yetersizliğinden dolayı uygulamada çökmeler ve donmalar meydana gelebilme olasılığı yükselecektir. Metraj hesaplamalarında yapı elemanlarının boyutları, koordinatları ve komşuluk ilişkilerini belirlemek için ve yapının 3B modelini oluşturmak için veritabanından çekilen veriler kullanılmaktadır.

3.1 Çizim Ara Yüzünün Oluşturulması

Tkinter, Python programlama dili için standart olarak kullanılan bir Grafiksel Kullanıcı Arayüzü (GUI) kütüphanesidir. Çapraz bir platform olan Tkinter Kütüphanesinin Windows işletim sistemi dışında macOS ve Unix gibi birçok işletim sisteminde çalışabilme özelliği güçlü ve sağlam ara yüzler oluşturma imkânı sağlamaktadır.

Tkinter kütüphanesindeki Tk() sınıfı ile geliştirilen programın ana penceresi oluşturulmaktadır. Oluşturulan ana pencere üzerinde çizim ara yüzü, butonlar ve diğer araçların yerleri belirlendikten sonra Frame() sınıfı kullanılarak araçlar gruplandırılmaktadır. Frame() sınıfı, farklı araçların ana penceredeki konumlarına göre düzenlenmesi ve gruplandırılması işlemlerinin gerçekleştirilmesi için çerçeve

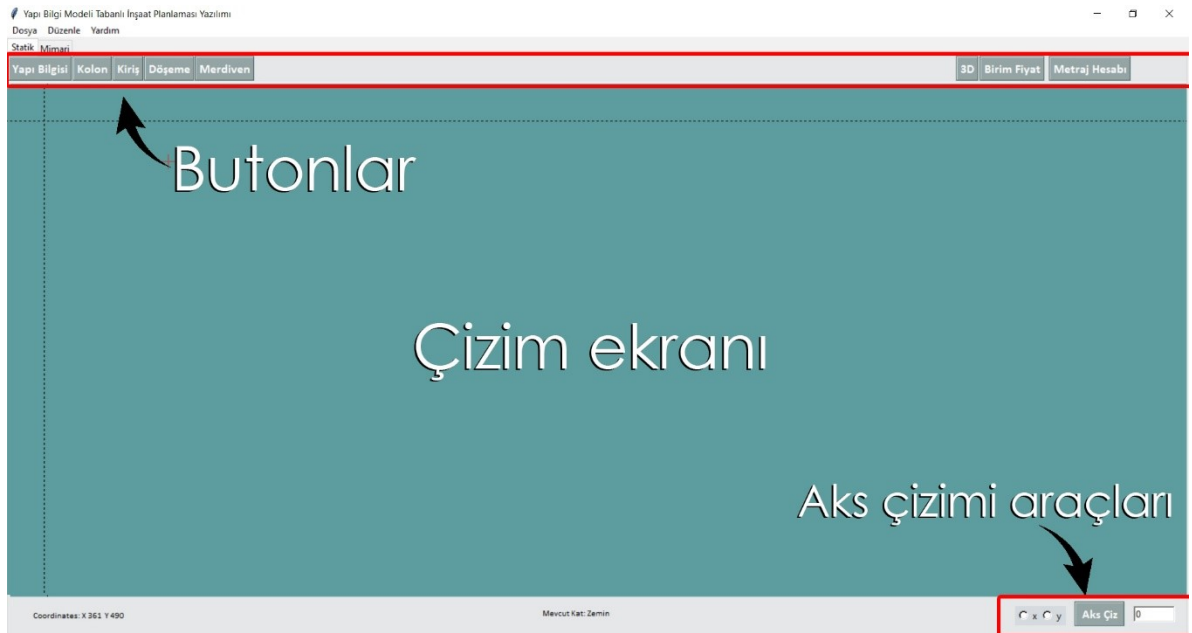
oluşturmaktadır. Şekil 3.2’de butonları bir arada tutan çerçevenin oluşturulmasını sağlayan kod parçacığı sunulmuştur. Tkinter kütüphanesinde geometri yönetimini gerçekleştirmek için pack(), grid() ve place() metotları kullanılmaktadır.

```
buttonFrame = Frame(tab1, width=1535, height=40, bd=0, bg="#E5E7E9")
buttonFrame.grid(row=0, column=0, sticky="nsew")
```

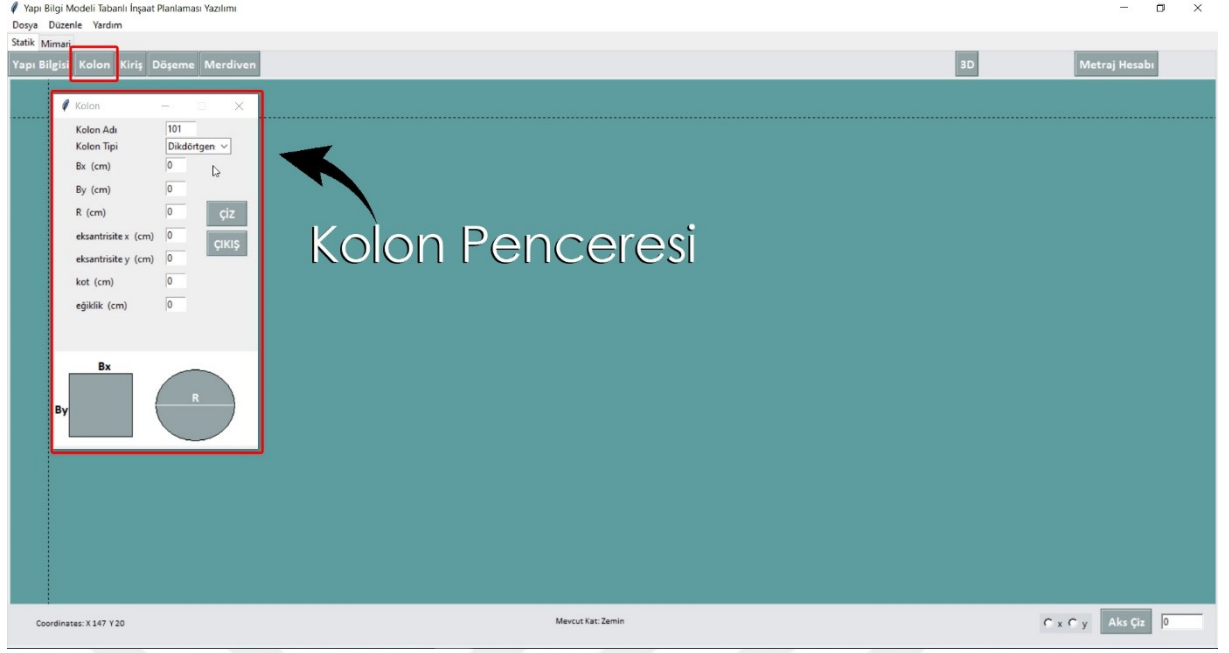
Şekil 3.2: Frame sınıfı kullanılarak çerçeve oluşturulması

Çizim ara yüzünü oluşturmak için Canvas() sınıfı kullanılmıştır. Canvas() sınıfı, üzerinde çizim yapılabilen tuval oluşturmak için kullanılmaktadır. Çizgi, dikdörtgen, çokgen, yay gibi birçok şekli çizmek için içerisinde metotlar bulundurmaktadır. Canvas() sınıfı ile oluşturulan çizim ara yüzü penceresinde ekran kaydırma (pan) işlemi yapılabilmesi için Canvas() sınıfı içerisindeki *scrollregion* parametresi kullanılmıştır.

Çizilecek olan yapı elemanlarının özelliklerinin kullanıcı tarafından girilmesini sağlamak için açılan (pop-up) pencereler, Toplevel() sınıfı ile oluşturulmuştur. Button() sınıfı kullanılarak butonlar oluşturulmuştur. Şekil 3.3’ te tıklandığında o elemana ait açılır pencereyi açan butonlar sunulmuştur.



Şekil 3.3: Kullanıcı Ara yüzü penceresinde butonların ve çizim ekranının konumları



Şekil 3.4: Kolon butonuna basıldığında yapı elemanlarından Kolona ait özelliklerin girilmesini sağlayan pencerenin açılması

Şekil 3.4’te sunulan kolon elemanına ait özelliklerin girilmesini sağlayan penceredeki metinler Label() sınıfı ile oluşturulmuştur. Label() sınıfı, resim ya da metinlerin pencere içerisinde görüntülenmesini sağlamaktadır. Kolon penceresindeki metin kutuları ise Entry() sınıfı ile oluşturulmuştur.

```
label_column_name = Label(column_data_popup, text="Kolon Adı")
label_column_name.place(relx=0.10, rely=0.01)

entry_column_name = Entry(column_data_popup, textvariable=kolon_ad, width=6)
entry_column_name.place(relx=0.55, rely=0.01, relwidth=0.15)
```

Şekil 3.5: Label ve Entry sınıflarının kullanımı

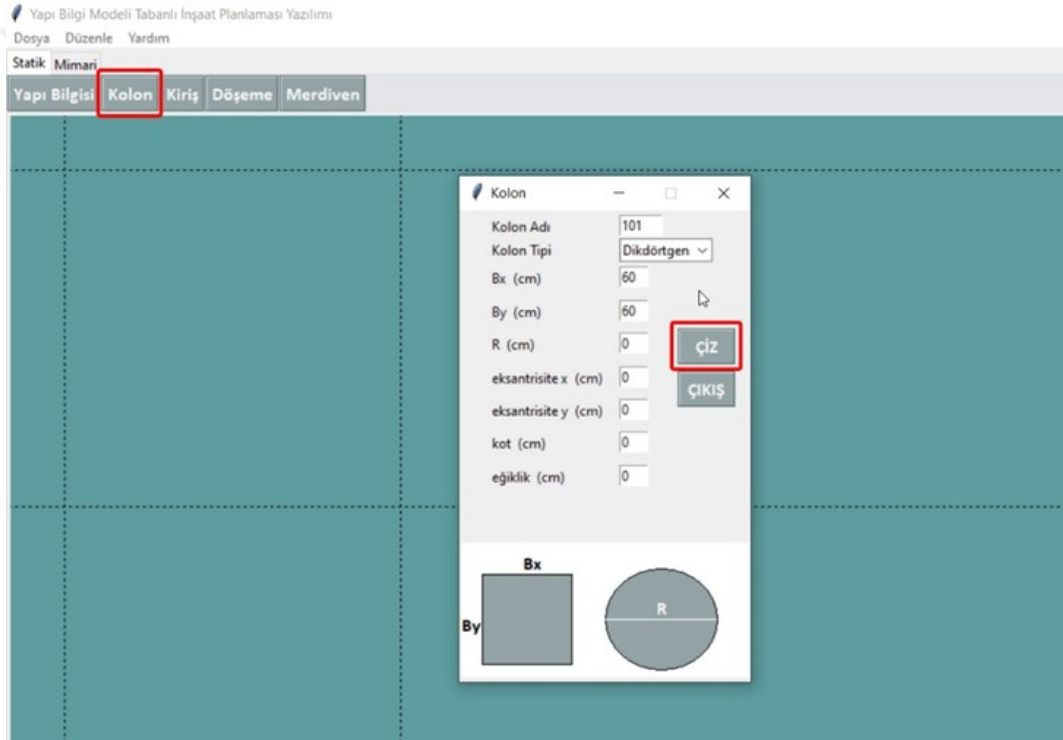
Entry() sınıfı, kullanıcı tarafından girilen veriler için girdi kutuları sağlamaktadır. Şekil 3.5’te kolon penceresine ait Label() ve Entry() sınıflarının kullanıldığı bir kod parçacığı sunulmuştur.

Entry() sınıfları ile oluşturulan girdi kutularındaki değerleri tutmak için değişkenlerden yararlanılır. Örneğin tam sayı tutmak için IntVar() sınıfı kullanılırken, dizi ya da karakter tutmak için StringVar() sınıfı kullanılmaktadır. Değişkenler aracılığıyla tutulan değerlerin çağırılması için get() metodundan yararlanılır. Örneğin Şekil 3.5’te “kolon_ad” değişkeni kolon_ad.get() komutu ile çağırılabilir.

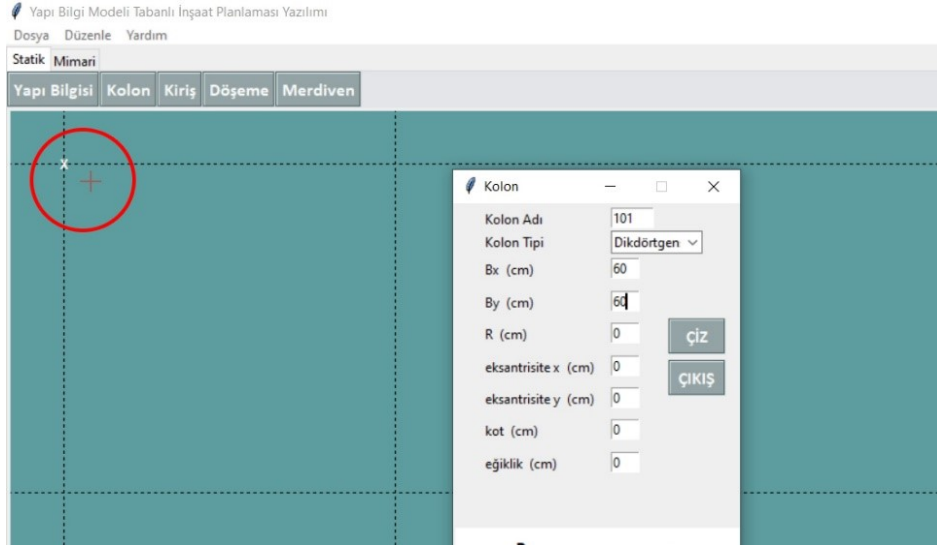
3.2 Çizim İşleminin Gerçekleştirilmesi

Bölüm 3.1’de çizim ara yüzünün oluşturulması ve çizilecek olan eleman ile ilgili özelliklerin girilebildiği pencerelerin oluşturulmasından bahsedilmiştir. Bu bölümde oluşturulan pencereler kullanılarak çizimin nasıl yapılacağı açıklanmaktadır.

Şekil 3.6’da gösterildiği gibi çizilecek olan eleman ile ilgili gerekli özellikler girildikten sonra “Çiz” butonuna basılır. Ardından çizim ekranı üzerindeki akslar yardımıyla elemanın çizileceği yere farenin sol tuşu ile tıklanarak çizim gerçekleştirilir. Sadece tek düğüm noktası ile ifade edilen kolon ve benzer yapısal elemanlarda çizimin gerçekleştirilmesi için tek fare tıklaması yeterli iken iki düğüm noktası ile tanımlanan kiriş için 2 farklı noktanın tıklanarak tanımlanması gerekmektedir.



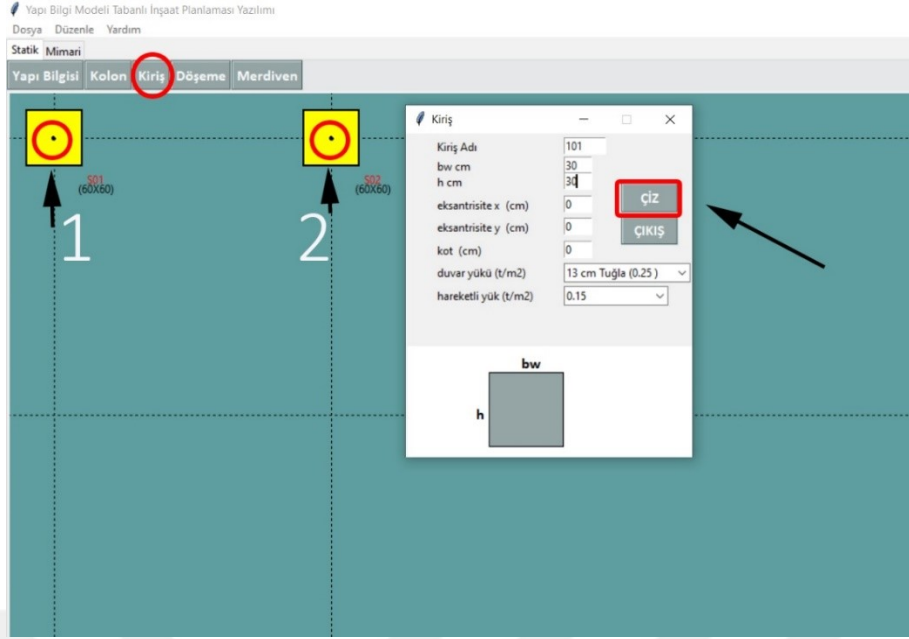
Şekil 3.6: Kolon penceresinde kolonun özelliklerinin girilmesinin ardından “ÇİZ” butonuna basılması



Şekil 3.7: Kolonun çiziminin yapılacağı aksların kesiştiği noktaya farenin sol tuşu ile tıklanması

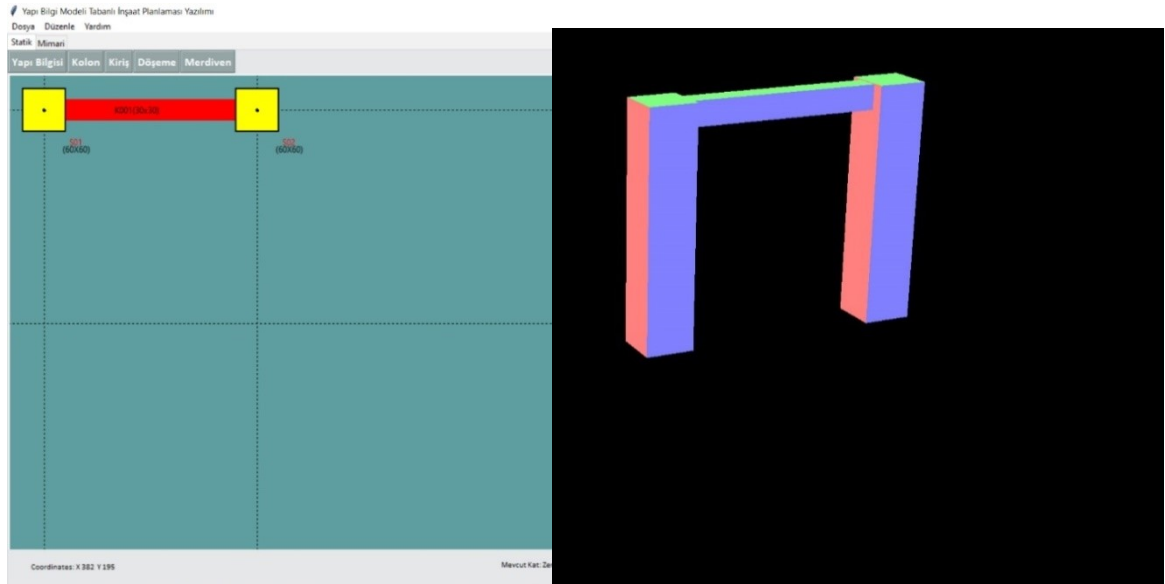


Şekil 3.8: Kolon çiziminin gerçekleştirilmesi



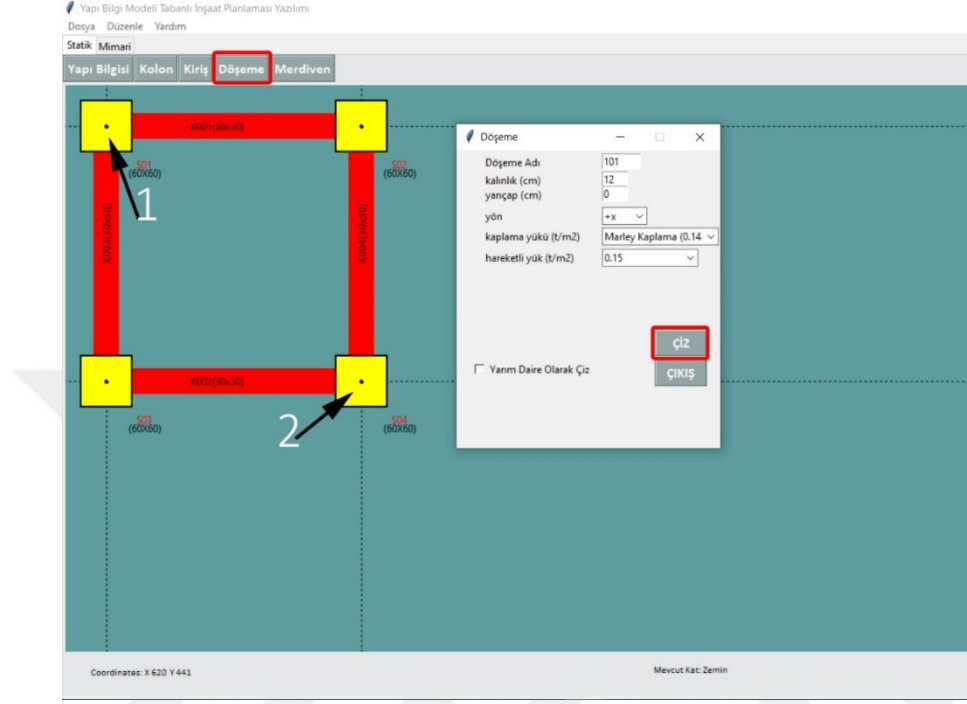
Şekil 3.9: Kiriş penceresinde kirişin özelliklerinin girilmesinin ardından “ÇİZ” butonuna basılması

Kiriş elemanını çizmek için “Kiriş” butonuna basıldıktan sonra açılan pencerede kirişe ait özellikler girilir ve “Çiz” butonuna basılır. Ardından Şekil 3.9’ da gösterildiği gibi sırasıyla 1 ve 2 numaralı düğüm noktalarına farenin sol tuşu ile basılır. Böylece kiriş elemanının çizimi gerçekleştirilir.

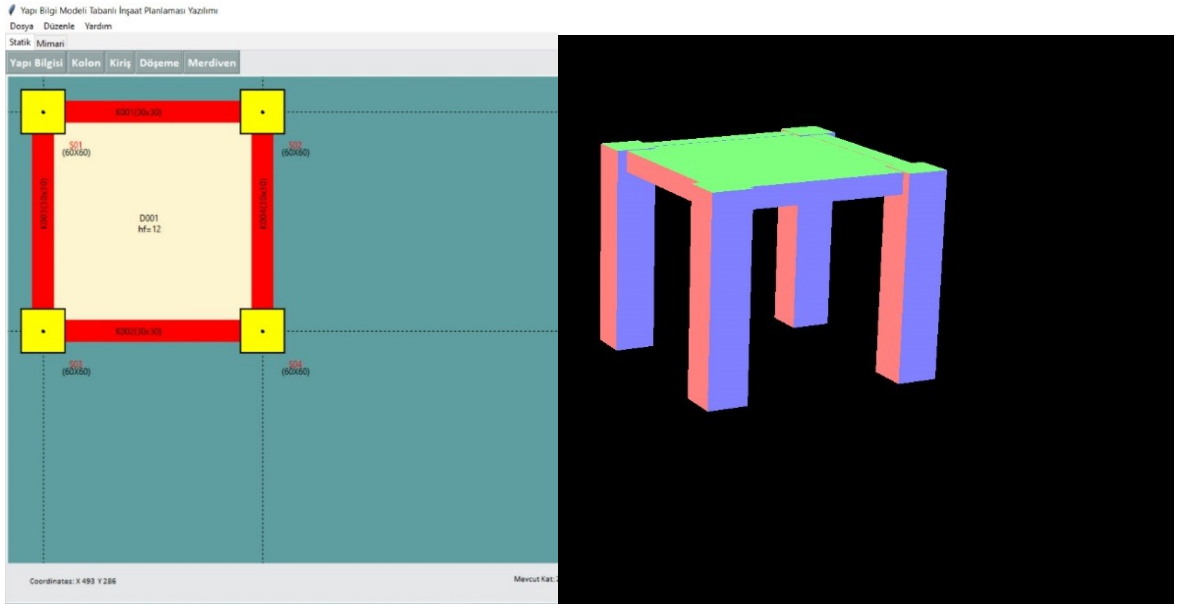


Şekil 3.10: Kiriş çiziminin gerçekleştirilmesi ve 3B Modelinin görüntülenmesi

Döşeme elemanını çizmek için “Döşeme” butonuna basıldıktan sonra açılan pencerede döşemeye ait özellikler girilir ve “Çiz” butonuna basılır. Ardından Şekil 3.10’da gösterildiği gibi sırasıyla 1 ve 2 numaralı düğüm noktalarına farenin sol tuşu ile basılır. Böylece döşeme elemanının çizimi gerçekleştirilir.



Şekil 3.11: Döşeme penceresinde döşeme özelliklerinin girilmesinin ardından “ÇİZ” butonuna basılması



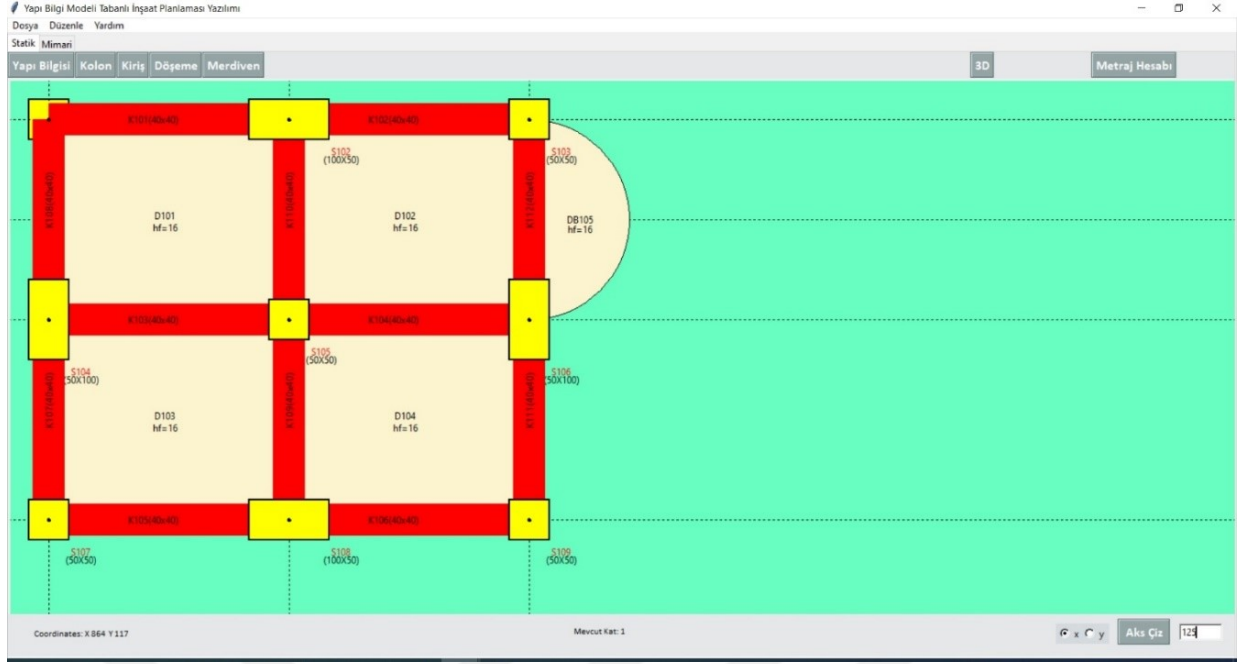
Şekil 3.12: Döşeme çiziminin gerçekleştirilmesi ve 3B Modelinin görüntülenmesi

“Çiz” butonuna basıldıktan sonra çizimi gerçekleştiren fonksiyon çağrılmaktadır. Python dilinde fonksiyon oluşturmak için fonksiyon adının önüne “def” anahtar sözcüğü kullanılır. Şekil 3.13’ de Python dilinde fonksiyon sözdizimi sunulmuştur. Şekil 3.10 ve Şekil 3.12 incelendiğinde çizim ekranındaki kullanıcı tarafından çizilen planların 3B modeli oyun motoru ve veritabanı kullanılarak otomatik bir şekilde kullanıcı müdahalesi olmadan oluşturulmaktadır. Bölüm 3.6’da yapının 3B görselleştirilmesi hakkında daha detaylı açıklama yapılmıştır.

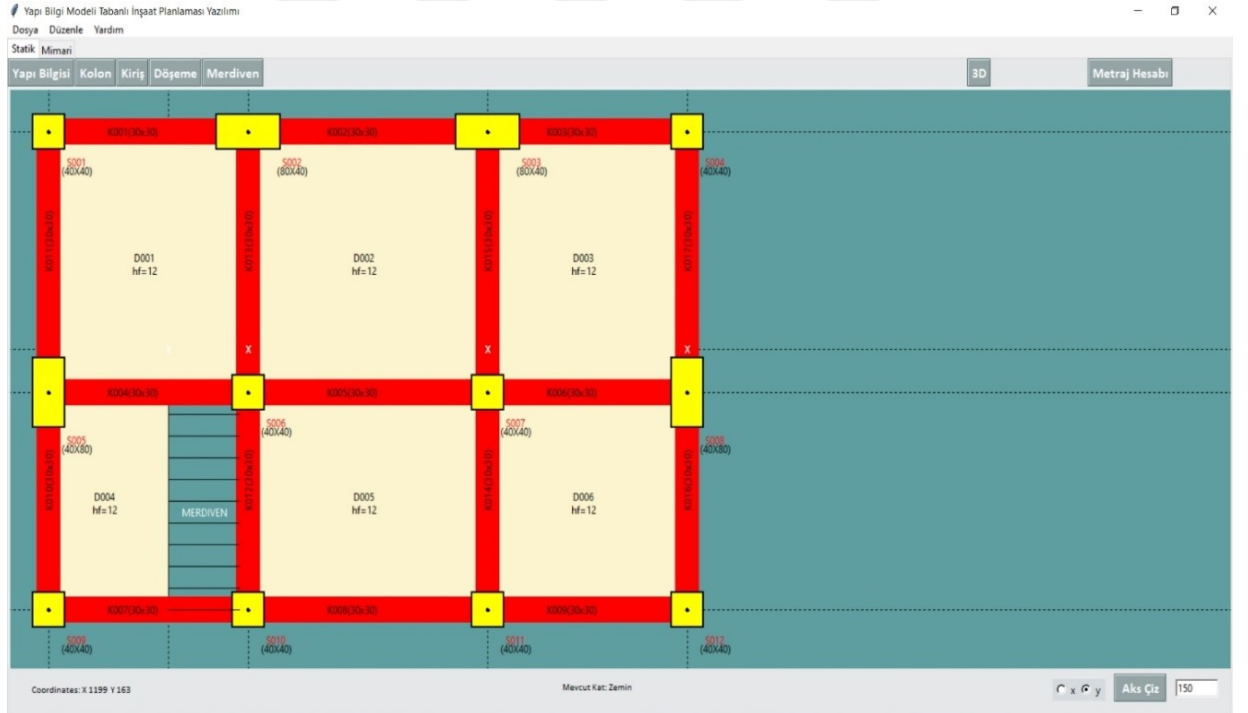
Çizilecek olan elemanın şekline göre Canvas() sınıfına ait metotlar kullanılır. Örneğin dikdörtgen kolon elemanının 2B gösterimi için create_rectangle() metodu kullanılırken yarım daire balkon için add_circle() metodu kullanılmıştır. Metotlar yalın haliyle değil elemana göre özelleştirilmiş haliyle kullanılmıştır. Çizim fonksiyonun fare tıklamaları ile gerçekleştirilmesi için bind() metodu kullanılmıştır. bind() metodu, belirli bir işlemin belirli bir pencerede fare ve klavye düğmelerini kullanarak gerçekleştirilmesini sağlamaktadır.

Çizimi gerçekleştirilen elemanlar birer nesne olarak Python’da oluşturulan bir liste içerisine eklenmektedir. Benzer elemanlar aynı liste içerisine eklenmektedir. Çizilen bir elemanı silmek için eleman üzerinde sağ tıklanarak çıkan menüden “Sil” seçeneği seçilerek eleman silinebilmektedir. 2B çizim ekranında silinen eleman ait olduğu liste içerisinde silinmektedir.

Çizilecek olan elemanların doğru noktaya konumlanması için fare ile düğüm noktasına yaklaşıldığında uyarı verilmektedir. Uyarı çıktığı anda çizimi gerçekleştirmek için farenin sol tuşuna basıldığında çizim tam olarak istenilen noktaya göre çizilecektir. Uyarı sistemi sayesinde çizimler daha az hata ile gerçekleştirilmekte ve hesaplamalar daha doğru sonuç vermektedir. Geliştirilen program ile çizilen A ve B yapılarının kalıp planları Şekil 3.13 ve Şekil 3.14’ de sunulmuştur. Kalıp planları verilen yapıların metraj değerleri Bölüm 4.1’ de hesaplanmıştır.



Şekil 3.13: 3 katlı A Yapısının Kalıp Planı Çizimi



Şekil 3.14: 2 katlı B Yapısının Kalıp Planı Çizimi

3.3 Veri Tabanının Oluşturulması ve Yönetilmesi

Geliştirilen programda veri tabanı oluşturmak için sqllite3 veri tabanı kullanılmıştır. SQLite açık kaynak kodlu veri tabanı motorudur. Yapılandırılmış verilerin saklandığı veri tabanları özellikle YBM sürecinde çok büyük bir öneme sahiptir. YBM tabanlı geliştirilen

programda çizilen yapı elemanlarının özellikleri, yapı bilgileri vb. özellikler sqlite3 ile oluşturulan veri tabanında kaydedilmektedir. Veri tabanına kaydedilen veriler daha sonra metraj hesaplamalarında ve yapının 3B modelinin oluşturulmasında kullanılmaktadır. Şekil 3.15'te B yapısına ait çizimi gerçekleştiren döşemelerin veri tabanı sunulmuştur. Döşemelerin başlangıç ve bitiş noktaları, döşeme kalınlığı gibi bilgiler veri tabanında yer almaktadır.

	id	Xbas	Ybas	Xbit	Ybit	hf	dosemeNo	dosemeAd	dosemeTipi	dosemeCapi	mevcutKat
	Filtre	Filtre	Filtre	Filtre	Filtre	Filtre	Filtre	Filtre	Filtre	Filtre	Filtre
1	1	50	50	300	350	12	2392	001	0	0	0
2	2	50	50	300	350	12	100	101	0	0	1
3	3	300	50	600	350	12	2479	002	0	0	0
4	4	300	50	600	350	12	103	102	0	0	1
5	5	600	50	850	350	12	2553	003	0	0	0
6	6	600	50	850	350	12	106	103	0	0	1
7	7	300	350	600	600	12	2621	005	0	0	0
8	8	300	350	600	600	12	109	105	0	0	1
9	9	600	350	850	600	12	2684	006	0	0	0
10	10	600	350	850	600	12	112	106	0	0	1
11	11	50	350	200	600	12	2878	004	0	0	0
12	12	50	350	300	600	12	115	104	0	0	1

Şekil 3.15: sqlite3 ile oluşturulan B yapısına ait döşemelerin veri tabanı

```
# ----- Kiriş Database -----
def beam_data():
    con_3 = sqlite3.connect("beam.db")
    cur_3 = con_3.cursor()
    cur_3.execute(
        "CREATE TABLE IF NOT EXISTS beam (id INTEGER PRIMARY KEY,\
        Gx text, Gy text, Genislik text, Derinlik text, Uzunluk text, Gxbas text, Gybas text, \
        Gxbit text, Gybit text, KirişEx text, KirişEy text, kirişNo text, kirişAd text, mevcutKat text)"
    )
    con_3.commit()
    con_3.close()
```

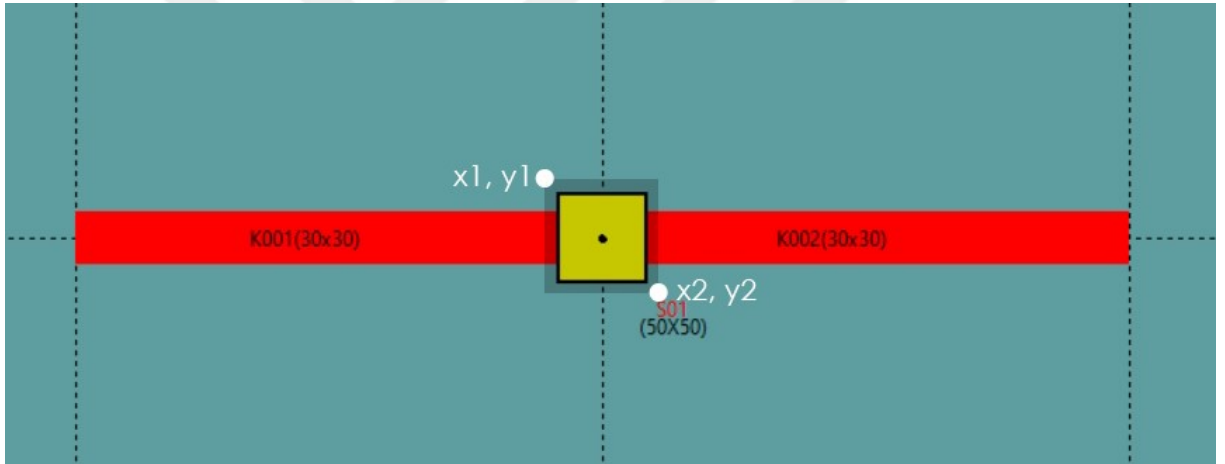
Şekil 3.16: sqlite3 ile veri tabanı oluşturma

Python sqlite3 ile veri tabanı oluşturmak için öncelikle connect() metodunu kullanarak sqlite3 ile etkileşime geçilmektedir. Ardından veri tabanında işlem yapabilmek

için `cursor()` metodu kullanılarak bir nesne oluşturulmaktadır. Veri tabanında yapılacak işlemler `execute()` metodu ile yerine getirilmektedir. Açılan her veri tabanı işlem tamamlandıktan sonra `close()` metodu ile kapatılmalıdır aksi takdirde bellekte yer kaplamaya devam edecektir. Şekil 3.16’ da kiriş elemanı için “beam.db” adında bir veri tabanı oluşturulmuştur. SQLite veri tabanı motorunun birçok işletim sisteminde (Windows, MacOS, Android vs.) çalışabilmesi çok büyük bir avantaj sağlamaktadır.

3.4 Çizilen Elemanların Komşuluklarının Tespit Edilmesi

Tkinter kütüphanesinin sınıflarından biri olan `Canvas()` sınıfına ait `find_overlapping()` ve `find_closest()` metotları yapı elemanlarının komşuluklarını tespit etmek için kullanılmıştır. Bu metotlar arasında en iyi sonucu veren metot yapı elemanının durumuna ve sonucun doğruluğuna göre tespit edilmiştir.



Şekil 3.17: S01 Kolonu ve Komşu Kirişler

Şekil 3.17’ de sunulan S01 kolonuna komşu olan elemanları tespit etmek için öncelikle kolona ait veri tabanından S01 kolonunun koordinat bilgileri çekilir. Elde edilen koordinat bilgileri doğrultusunda S01 kolonuna komşu olan elemanları tespit etmek için kolonun ebatlarından biraz daha büyük bir alan oluşturulur. S01 kolonuna ait komşu elemanları tespit edebilmek için $(x1, y1)$ ve $(x2, y2)$ noktalarından geçen bir dikdörtgen alan oluşturulmuştur. Ardından “`find_overlapping(x1, y1, x2, y2)`” metoduna ilgili parametreler girilerek oluşturulan dikdörtgen alanı içersinde arama yapılır. Arama sonuçlarında birden çok eleman tespit edilebilir. Bu yüzden istenilen elemanı tespit edebilmek için çizim aşamasında çizilen elemanlara “tag” parametresi verilmektedir.

Örneğin kirişler için “tag” parametresi “tag=kiris” olarak verilmiştir. “tag” parametresine göre karşılaştırma yapılarak komşu elemanların tipi tespit edilmektedir.

3.5 Metraj ve Maliyet Hesabı

Şekil 3.15’te sunulan veritabanından her yapı elemanı için ayrı ayrı oluşturulmuştur. Veritabanındaki depolanmış verilerin yapının metraj hesaplamalarında nasıl kullanıldığı bu bölüme ait alt başlıklarda anlatılmaktadır.

3.5.1 Kalıp metrajı

Kolon yapı elemanı için elemana ait veritabanından en, boy ve dışmerkezlik verileri ile kat yükseliği verisi alındıktan sonra her bir kolon için yanal alan hesaplanır. Ardından veritabanından her bir elemanın koordinat bilgileri çekilir. Alınan bu koordinat bilgileri Bölüm 3.4’te anlatıldığı gibi elemanın komşularını tespit etmek için kullanılır. Komşu elemanların (döşeme, kiriş) kolona temas eden yüzeylerinin alanı hesaplanarak (minha) kolonun yanal alanında çıkarılır. Böylece kolon için kalıp metrajı hesaplanmış olur.

Kiriş yapı elemanı için elemana ait veriler veritabanından çekilir. Veritabanından çekilen koordinat bilgileri Bölüm 3.4’ te açıklanan sınıflara ait metotların parametresi olarak tanımlanır ve kirişin mesnetleri tespit edilir. Böylece net açıklık mesafesi bulunur. Kirişin yanal alanı hesaplanır. Komşu elemanların kiriş ile temas eden yüzeylerin alanı (minha) hesaplanır. Ardından kirişin yanal alanından minhalar çıkarılarak kiriş kalıp metrajı hesaplanır. Kolon ve kirişin kalıp metrajını hesaplamak için anlatılan yöntem döşeme elemanı için de geçerlidir.

İç içe *for* döngüsü kullanılarak yapıdaki her bir kat için o kata ait tüm yapı elemanlarının kalıp metrajı yukarıda anlatılan yöntem uygulanarak hesaplanmıştır. Her bir yapı elemanı tipi için kalıp metrajı değerleri ayrı ayrı değişkenlere atanmıştır. Her bir değişkendeki değerler toplanarak toplam kalıp metrajı hesaplanmıştır. Hesaplanan toplam kalıp metrajı değeri ile kalıp birim fiyatı çarpılarak maliyeti hesaplanmıştır.

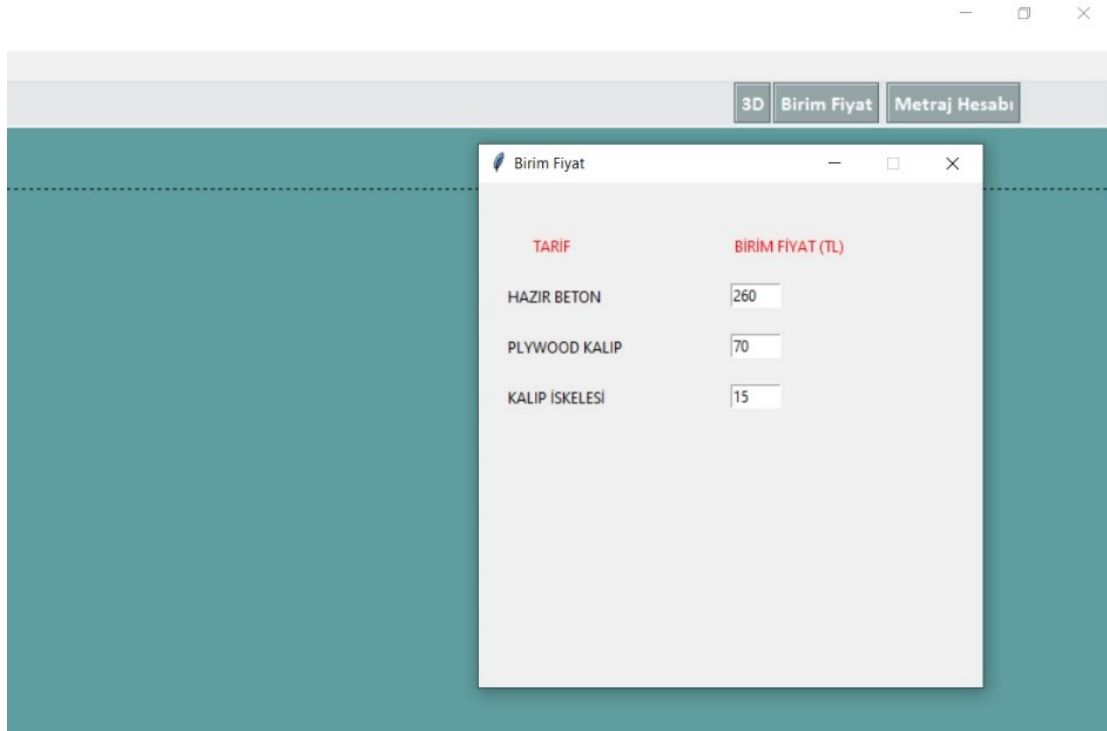
3.5.2 Beton metrajı

Beton metrajı hesabı için ilgili elemanların ebatları veri tabanından çekilir ve her eleman için genişlik, uzunluk ve yükseklik değerleri çarpılarak elemanın hacmi hesaplanır. Böylece beton metrajı elde edilir. Kalıp metrajında olduğu gibi iç içe *for* döngüsü kullanılarak her bir eleman tipi ve kat için beton metrajı değeri elde edilir. Her bir eleman tipine ait beton metrajı değeri toplanarak toplam beton metrajı hesaplanmıştır. Hesaplanan toplam beton metrajı değeri ile beton birim fiyatı çarpılarak maliyeti hesaplanmıştır.

3.5.3 İskele metraji

İskele metraji için metraji hesaplanacak olan katın yüzey alanı ile kat yüksekliği çarpılır. Çarpım sonucundan ilgili katın beton metraji çıkartılarak her kat için ayrı ayrı olacak şekilde iskele metraji değerleri hesaplanmış olur.

Geliştirilen programda Şekil 3.18’de gösterildiği gibi “Birim Fiyat” butonuna basılarak ilgili tarifler için rayiç bedelleri kullanıcı tarafından girilir. Ardından “Metraj Hesabı” butonuna basılarak metraj ve maliyet hesabı yapılır. Sonuçlar Rapor şeklinde yeni bir pencere açılarak kullanıcıya sunulur.



TARİF	BİRİM FİYAT (TL)
HAZIR BETON	260
PLYWOOD KALIP	70
KALIP İSKELESİ	15

Şekil 3.18: Birim Fiyat Penceresi

3.5.4 Donatı Metraji

Bu çalışmada yapı elemanlarının donatı metraji statik ve betonarme hesap yöntemlerine girilmeden hesaplanmıştır. Donatı metraji için gerekli olan statik hesap verileri yardımcı programlardan ya da el hesabı sonucu elde edilen verilerden alınmıştır.

Geliştirilen yazılım, donatı metrajını hesaplarken elemanın anlamsal (semantik) modelini oluşturarak donatının yapı elemanı için uygunluk durumları test edilmiştir. Anlamsal modeller oluşturularak YBM modellerinin işlevselliği büyük oranlarda

iyileşmektedir (Sacks vd., 2017). Örneğin kiriş elemanı için boyuna donatıların net aralığı (c'_c) TS 500:2000 tarafından şu şekilde sınırlandırılmıştır:

1. $c'_c \geq 2.5 \text{ cm}$
2. $c'_c \geq \emptyset$ (donatı çapı)
3. $c'_c \geq (4/3)D_{\text{maks}}$ (maksimum agrega tane çapı)

Sınırlandırmalar dikkate alındığında D_{maks} ile ilgili olan sınırlandırmanın genellikle ihmal edildiği görülmektedir. Statik ve Betonarme hesap yapan programlarda elemana ait özellikler girilirken malzeme kısmında beton sınıfı istenirken genellikle D_{maks} istenmemektedir. Bir yapının tasarımında D_{maks} değerine statiker karar vermektedir. Geliştirilen uygulamada malzeme kısmında beton sınıfı dışında D_{maks} değeri de kullanıcıdan istenmektedir. Bu sayede donatı net aralığı için yukarıda verilen sınır değerlerden D_{maks} ile ilgili olan sınırlandıma metraj hesabında uygulanmaktadır.

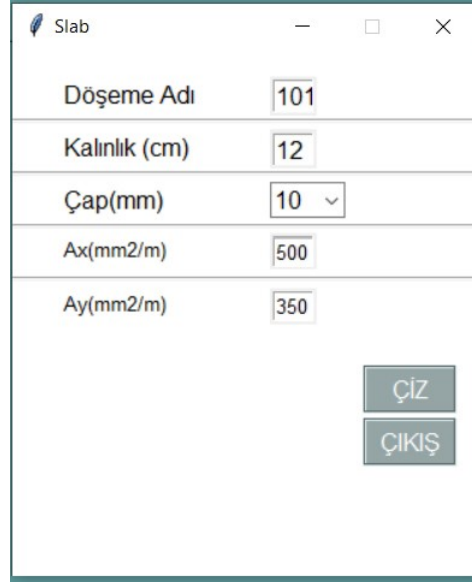
“ $c'_c \geq (4/3)D_{\text{maks}}$ ” şartının sağlanmadığı durumlarda eleman için yönetmeliklerdeki sınır değerler dikkate alınarak yeni bir çap belirlenmektedir ve yönetmeliklerdeki sınırlandırmalar tekrar yapılmaktadır.

Döşemenin çalışan doğrultu sayısına göre davranışı, donatı düzeni ve yönetmelikler tarafından belirlenen donatı aralığı sınır değerleri farklıdır. Donatı aralığı, düz donatıdan düz donatıya ya da pilye donatısından pilye donatısına kadar olan mesafeyi ifade etmektedir. Döşemenin iki doğrultuda mı yoksa tek doğrultuda mı çalıştığı belirlendikten sonra sınır değerler kontrolü yapılmaktadır. Ardından donatı aralıkları belirlenmektedir.

Döşemenin her bir kenarında süreklilik analizi yapabilmek için öncelikle döşemenin her bir kenarına ait komşu elemanlarını tespit etmek gerekmektedir. Kirişli döşemenin herhangi bir kenarının bir kirişe veya perdeye oturuyor olması ve komşu döşemeninde ortak kenarı olması durumunda ilgili kenar “sürekli kenar” olarak adlandırılmaktadır. Geliştirilen yazılımda “sürekli kenar” durumu “1” olarak nitelendirilmektedir. Kirişli döşemenin herhangi bir kenarının bir kirişe oturuyor olması fakat başka bir döşeme ile komşuluk ilişkisinin bulunmaması durumunda ilgili kenar “süreksiz kenar” olarak adlandırılmaktadır. Geliştirilen yazılımda “süreksiz kenar” durumu “2” olarak nitelendirilmektedir. Kenarın hiçbir yere oturmaması durumunda ise “kenar boşta” durumu ortaya çıkmaktadır. Bu durum yazılımda “0” olarak nitelendirilmektedir. Geliştirilen yazılımda kenarın sürekli olup olmadığını tespit etmek için tkinter kütüphanesi içerisindeki “Canvas()” sınıfına ait “find_overlapping()” metodundan yararlanılmıştır. Çizimin

yapıldığı tuval nesnesinin üzerindeki belirli bir koordinat değerleri aralığındaki bölgenin içerisinde kalan şekillerin listesini döndürmeye yaramaktadır. Örneğin döşeme, kolon, kiriş gibi şekillerin çizildiği tuval nesnesi için “find_overlapping(0, 0, 100, 100)” metodu kullanıldığı durumda metodun içerisinde verilen parametrelerin oluşturduğu n1(0, 0) ve n2(100, 100) noktaları arasında kalan bölge içerisindeki şekillerin listesi döndürülmektedir. Bu durumda her bir kenarın bulunduğu bölgenin koordinatları ilgili metodun parametresi olarak girildiğinde kenara olan komşu elemanların listesi döndürülmüş olur. Bu durumda birçok nesne komşu elemanlar listesinde bulunabilir. İstenilen elemanları komşu elemanların bulunduğu listeden çekebilmek için “tag” parametresi kullanılmaktadır. Çizilen her bir elemanın “tag” parametresi kendisine özgüdür. Örneğin kolon elemanı için “kolon”, kiriş elemanı için “kiris” ve döşeme elemanı için “doseme” tag parametreleri kullanılmıştır. Böylece ilgili kenara ait komşu elemanlar listesindeki elemanların hangi tipte bir eleman olduğu anlaşılmaktadır. Komşu elemanlar listesinde o kenara komşu olan bir kiriş ve kirişin devamında bir döşeme tespit edildiği durumda süreklilik durumunu gösteren veri listesinde ilgili kenar için “1” yazılmaktadır. Örneğin her bir kenarı sürekli olan döşeme için oluşturulan süreklilik durumunu gösteren veri listesi “[1, 1, 1, 1]” şeklindedir.

Döşeme donatı metrajı için gerçekleştirilen vaka çalışmalarında kirişli döşeme tipi kullanılmıştır. Kirişli döşeme tipi deprem anındaki davranışı en iyi olan döşeme tipidir (Topçu, 2019). “Döşeme” butonuna bastıktan sonra açılan pencerede döşemenin adı, kalınlığı, donatı çapı, x ve y yönleri için 1000 mm’ lik şeritteki donatı alanı değerleri kullanıcı tarafından girilmektedir. Şekil 3.19’da döşeme bilgilerinin girildiği pencere gösterilmektedir.



Döşeme Adı	101
Kalınlık (cm)	12
Çap(mm)	10
Ax(mm2/m)	500
Ay(mm2/m)	350

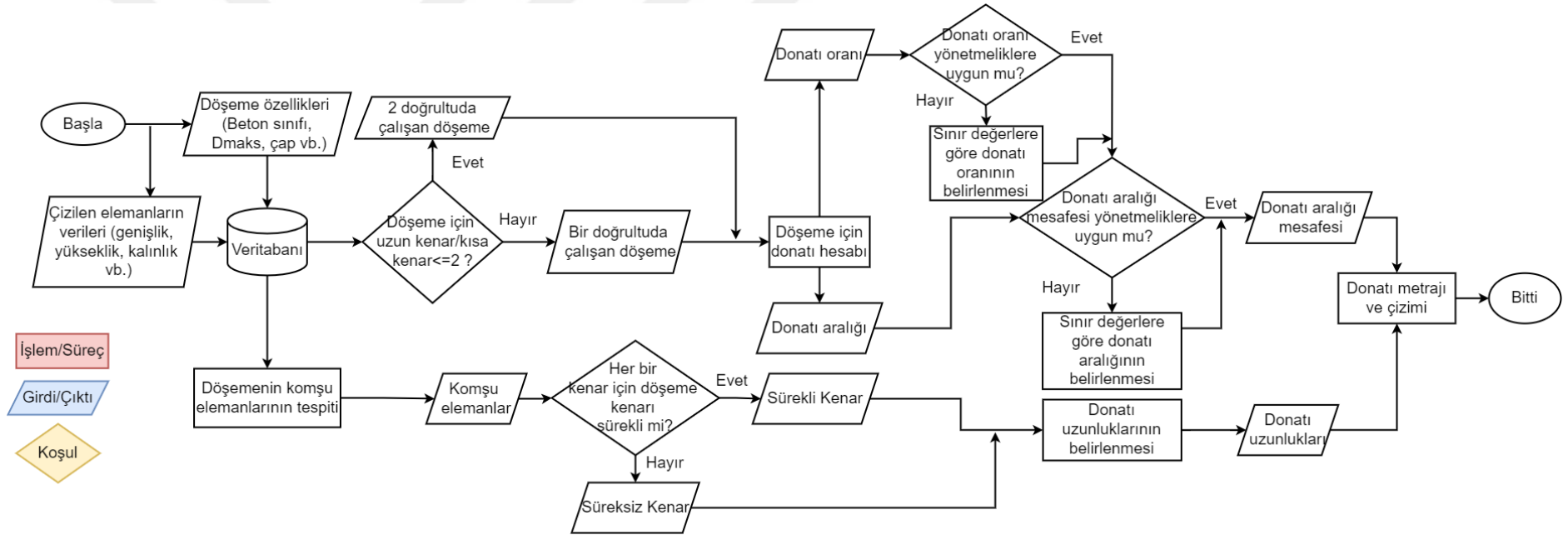
ÇİZ

ÇIKIŞ

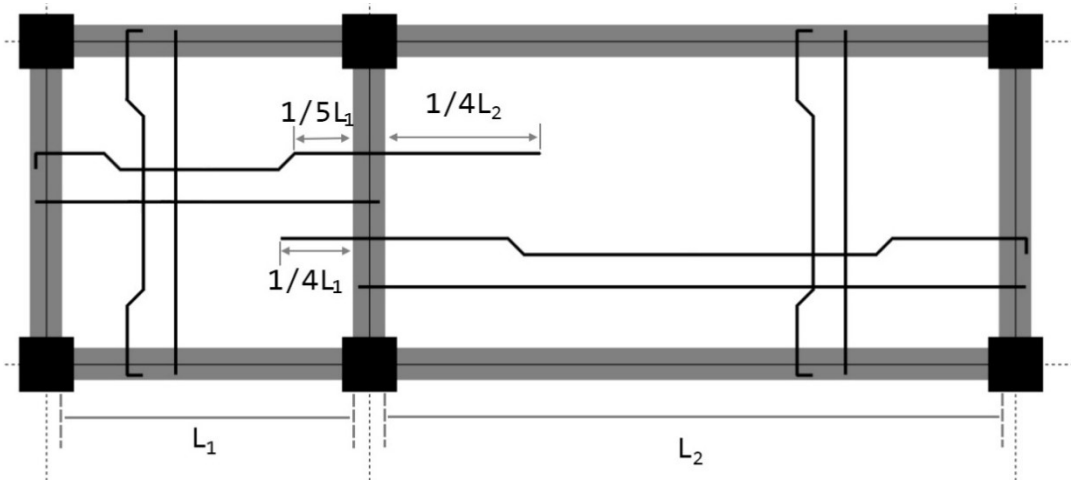
Şekil 3.19: Döşemeye ait verilerin girildiği pencere

Aksların ve yapı elemanlarının çizimi sonucu elde edilen yükseklik, genişlik gibi veriler ile döşemenin özellikleri sqlite veri tabanı motoru kullanılarak oluşturulan veritabanına aktarılmaktadır. Döşeme ile ilişkili veriler “slab.db” veritabanında saklanmaktadır. Çizim işlemi sona erdikten sonra kullanıcı arayüzündeki “Slab Analysis” butonuna basılarak döşeme analizi gerçekleştirilir. Şekil 3.20’ de gösterildiği gibi analiz işleminde önce kirişli döşemenin kaç doğrultuda çalışmakta olduğu tespit edilmektedir. Python programlama dilindeki “if-elif-else” koşul ifadesi kullanılarak döşemenin doğrultu sayısı tespit edilmektedir.

Döşeme kenarlarının süreklilik durumunun tespiti sonucunda döşeme donatılarının uzunlukları hesaplanmaktadır. Sürekli kenarlar için pilye donatıları komşu döşemenin net açıklığının $\frac{1}{4}$ ü kadar uzatılmaktadır. Şekil 3.21’ de iki doğrultuda çalışan döşeme için donatı detayı sunulmaktadır.



Şekil 3.20: Döşeme donatı metrajı hesabı akış şeması



Şekil 3.21: İki doğrultuda çalışan döşemenin sürekli kenarı boyunca pilye donatılarının uzatılması

Döşemenin donatı uzunlukları ve aralıkları tespit edildikten sonra düz ve pilye donatıların sayısı tespit edilmektedir. Düz donatıların sayısını tespit etmek için düz donatının doğrultusuna dik olan net açıklık düz donatı aralığına bölünür. Çıkan sonuç tam sayı ise pilye donatısı düz donatı sayısının bir eksiği olarak hesaplanır. Sonucun tam sayı olmadığı durumlarda ise öncelikle çıkan sonuç tam sayıya aşağı yuvarlanır. Elde edilen sayı ile donatı aralığı mesafesi çarpılarak net açıklıktan çıkarılır. Sonuç düz donatı aralığının yarısından küçükse düz donatı sayısı elde edilen sayının bir fazlasına pilye ise elde edilen sayıya eşit olur. Çıkarma işlemi sonucu elde edilen değer donatı aralığının yarısından büyükse düz donatı sayısı aralık sayısının bir fazlası ve pilye donatı sayısı da aralık sayısının bir fazlası olarak hesaplanır.

Donatı çapı, adedi, uzunluğu ve birim ağırlığı çarpılarak donatı metrajı hesaplanmaktadır. “Canvas()” sınıfı içerisindeki “create_line()” metodu kullanılarak düz ve pilye donatıların çizimi yapılmakta olup kullanıcıya açılır pencerede metraj tablosu ile birlikte sunulur.

Kiriş donatı metrajı TS500 ve TBDY 2018 yönetmeliğindeki sınırlamalar dikkate alınarak hesaplanmıştır. Özellikle TS500 tarafından zorunlu bir koşul olan “iki boyuna donatı arasındaki net donatı aralığının $4/3 D_{maks}$ değerinden küçük olması” şartı da dikkate alınarak metraj hesaplanmıştır. Günümüzdeki yazılımların elemanın malzeme özelliklerini dikkate alırken genellikle D_{maks} ile ilgili olan sınırlandırmaları dikkate

almamaktadır. Çünkü tasarım öncesi “Dmaks” değeri istenmemektedir. Fakat proje mühendisi beton sınıfı ile birlikte Dmaks değerini de belirlemek zorundadır. Geliştirilen yazılımda kirişin ve diğer elemanların beton sınıfı özelliği dışında Dmaks değeri de kullanıcı tarafından alınmaktadır.

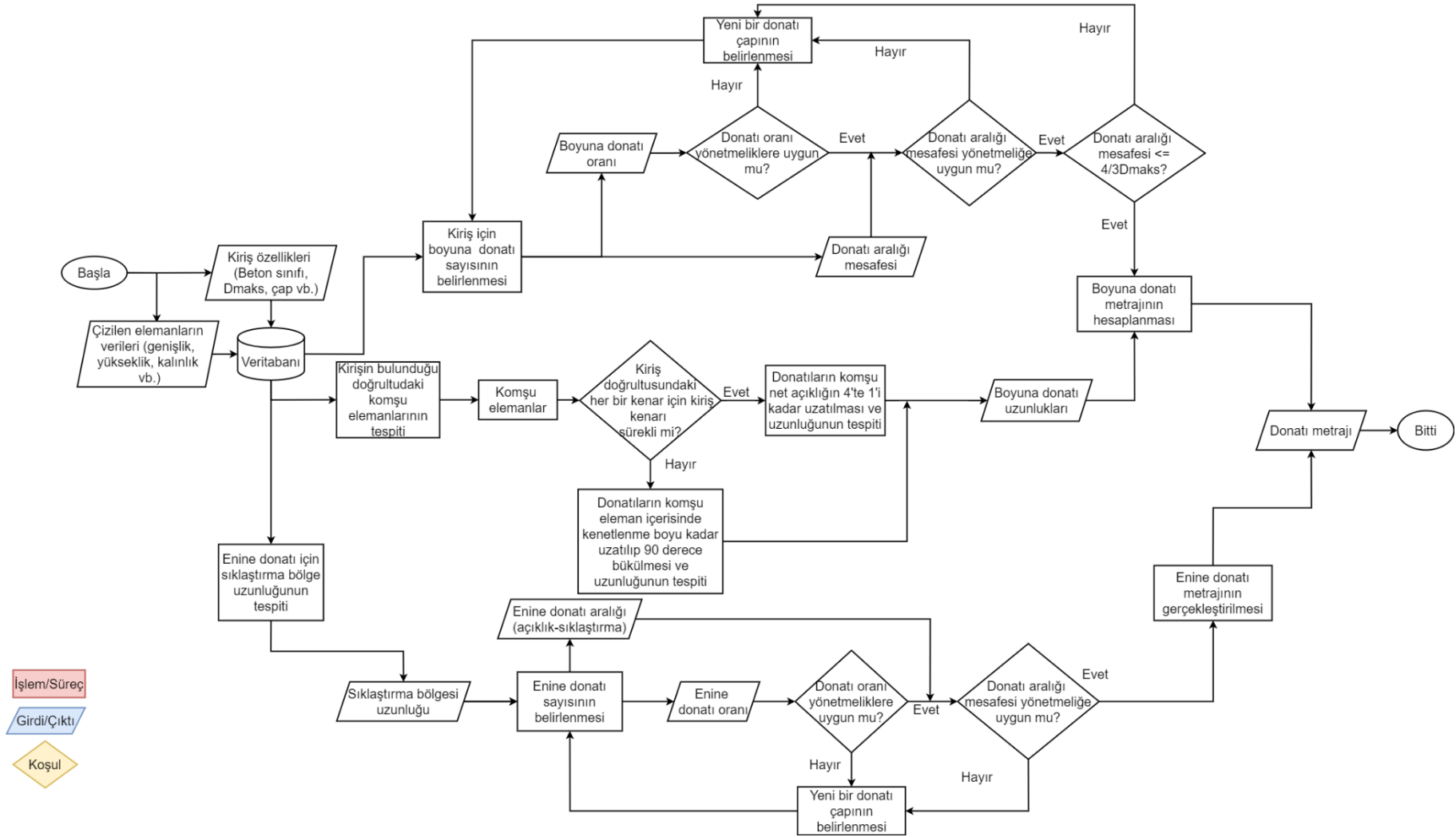
Şekil 3.22’ de kirişin donatı metrajı hesabı algoritmasının akış şeması sunulmuştur. Kirişin komşuluk ilişkileri Bölüm 3.5.4.1’de döşemenin komşuluk ilişkilerinin tespit edilme yöntemi ile aynı şekilde tespit edilmiştir.

Enine donatılar için sarılma (sıklaştırma) bölgesi uzunluğu kirişin yüksekliğinin 2 katı kadar alınmıştır. Eğer iki sarılma bölgesinin toplam uzunluğu kirişin net açıklığını geçiyorsa bu durumda kiriş boydan boya sıklaştırılmaktadır. Bu durumun kodlanmaması durumunda açıklık bölgesi uzunluğu negatif değerler alabilecek ve böylece yanlış bir hesap yapılmış olacaktır. Geliştirilen yazılım boyunca böyle durumların ortaya çıkması da değerlendirilmektedir. Fakat her yazılım için hata durumu olasılıklarının hepsinin tespit edilip düzeltilmesi mümkün değildir. Yazılım kullanıldıkça ve farklı yapı elemanları ve planları ile test edildikçe hatalar tespit edilip düzeltilebilmektedir.

- Net donatı aralığı (c'_c) ≥ 2.5 cm
- Net donatı aralığı (c'_c) $\geq \emptyset$ (donatı çapı)
- Net donatı aralığı (c'_c) $\geq (4/3)D_{maks}$ (maksimum agrega tane çapı)

Boyuna donatılar için yukarıda verilen durumlar tek tek uygulanmıştır. Özellikle beton dökümü esnasında agreganın donatı arasından geçememesi gibi durumların önüne geçilmesi sağlanmıştır.

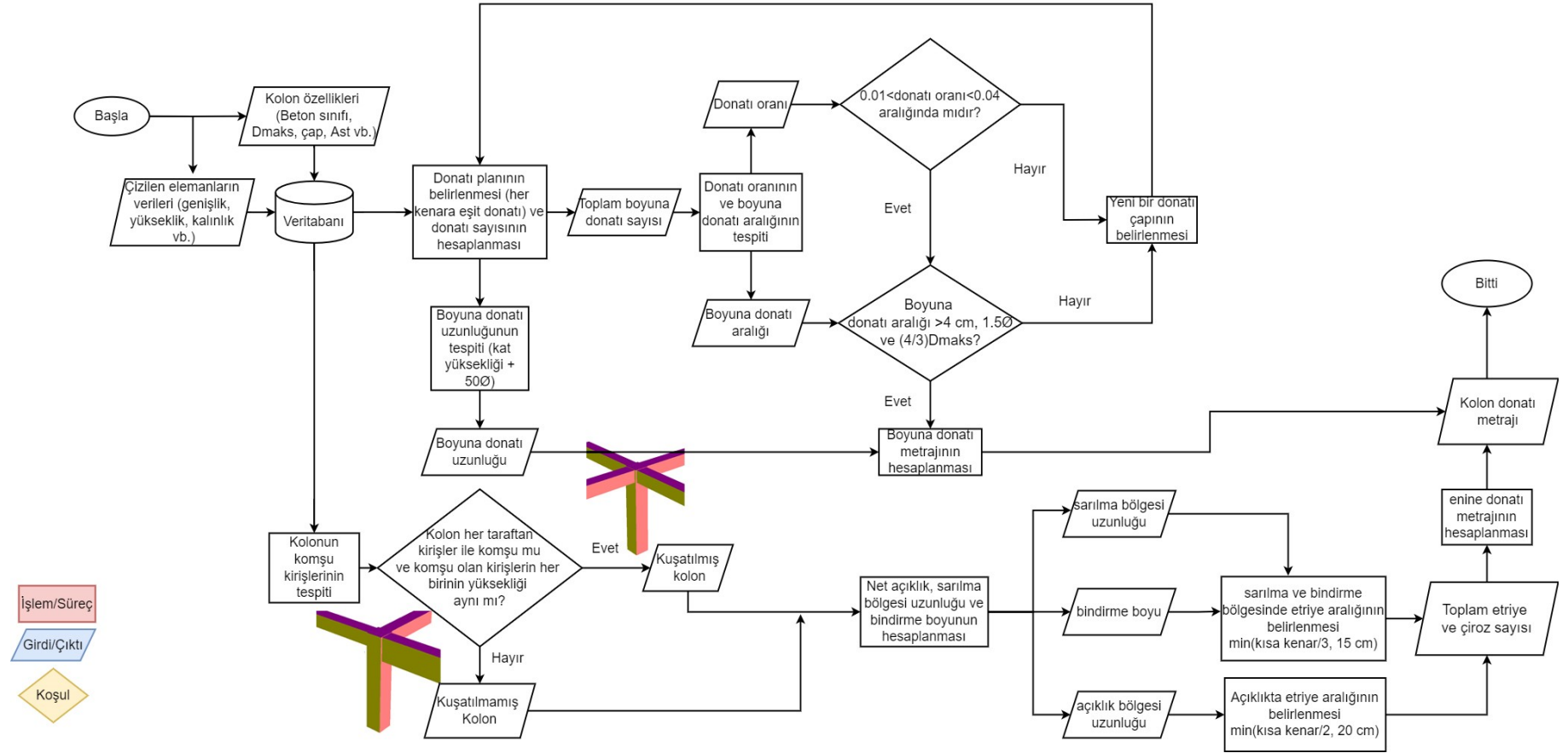
Pilye donatıları komşu kirişin net açıklığının $\frac{1}{4}$ ’ü kadar komşu kirişlere uzatılmıştır. Donatıların komşu kirişlere devam ettirilmediği durumlarda donatının mesnet içindeki kısmı 90 derece bükülmüştür. Kanca uzunluğu “ $12 \times \emptyset$ ” olarak hesaplanmıştır. Kirişlerde konstrüktif sargı donatıları kullanılmıştır.



Şekil 3.22: Kiriş donatı metrajı algoritmasının akış şeması

Kolon donatı metrajını hesaplamak için öncelikle kullanıcıdan donatı alanı (Ast) ile birlikte kolonun özellikleri alınır. Çizim arayüzü vasıtasıyla çizilen elemanların özellikleri ve kolonun özellikleri veritabanına akarılır. Kolonun ebatları, donatı çapı ve beton sınıfı vb. gibi özellikleri “column.db” adında veritabanında saklanmaktadır. Ast değeri ve boyuna donatı çapı veri tabanından çekilerek gerekli olan donatı sayısı bulunur. Ardından donatı oranı ve donatı aralığı kontrolleri yapılır. TBDY 2018 tarafından belirlenen sınır değerlerin aşılması durumunda yeni bir donatı çapı belirlenir ve donatı sayısı yeniden hesaplanmaktadır. Sınır değerlerin sağlanması halinde boyuna donatı metrajına geçilir. Kolonlar için minimum boyuna donatı çapı 14 mm olduğu için kullanıcı tarafından açılan pencerede seçilebilecek minimum boyuna donatı çapı 14 mm’ dir. Bu sayede yönetmeliğin belirlediği değerin altında bir donatı çapı girilmesi engellenmektedir. Aynı şekilde minimum etriye çapı 8 mm ($\emptyset_w$) olarak belirlenmiştir.

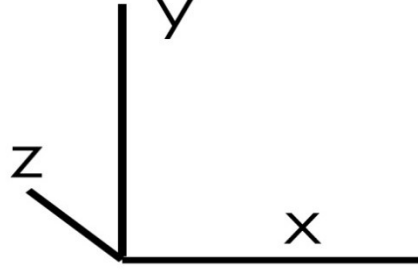
Enine donatıların metrajı için öncelikle kolonun net açıklığı, sarılma bölgesi uzunluğu ve bindirme bölgesi uzunluğu değerleri gereklidir. Net açıklığın tespiti için öncelikle kolonun kuşatılma durumu tespit edilir. Kolonun komşu kirişleri tespit edilerek her kenardaki komşu kirişin yüksekliğinin aynı olması durumunda kuşatılmış kolon, farklı olması durumunda kuşatılmamış kolon olarak adlandırılır. Bu durumda net açıklık (L_n) kat yüksekliğinden komşu kirişlerin yükseklikleri arasındaki en büyük yükseklik değeri çıkartılarak tespit edilir. Sarılma bölgesi uzunluğu kolonun uzun kenarının 1.5 katı, $L_n/6$ ve 50 cm değerlerinden en büyüğü alınarak belirlenmiştir. Bindirme boyu ise $50\emptyset$ alınmıştır. Etriye adım aralıkları yönetmeliğin belirlediği sınır değerlere göre açıklık, sarılma ve bindirme bölgeleri için belirlenmiştir. Açıklık için etriye aralığı kısa kenar/2 ve 20 cm değerlerinden küçüğü olarak belirlenirken sarılma bölgesi için kısa kenar/3 ve 15 cm değerlerinden küçüğü olarak belirlenmiştir. Etriyelerde yanal hareketi kısıtlamak için bombelenme boyu $25\emptyset_w$ olarak belirlenmiştir ve her bombelenme boyu mesafesinde bir çiroz atılmıştır. Kuşatılmış kolonlarda kuşatılmış bölgedeki etriye aralığı 15 cm alınırken kuşatılmamış bölge için 10 cm alınmıştır. Algoritmanın daha detaylı hali Şekil 3.23’ te sunulmuştur.



Şekil 3.23: Kolon donatı metrajı algoritmasının akış şeması

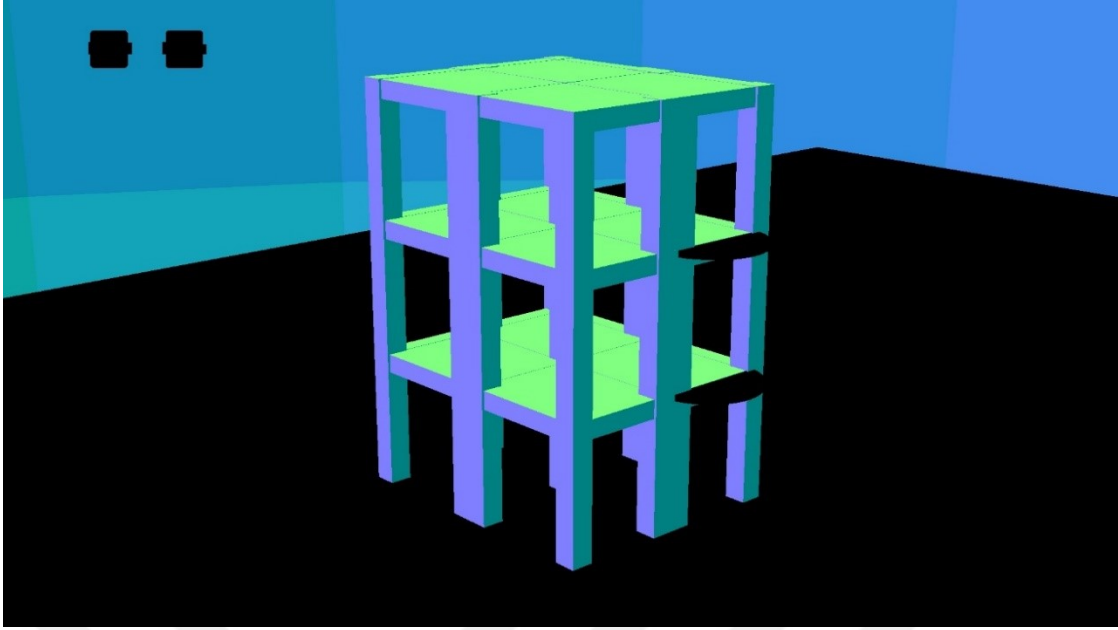
3.6 Yapının 3 Boyutlu Görselleştirilmesi

Geliştirilen program ile kalıp planı çizilen bir yapının 3 Boyutlu Modelini oluşturmak için Python programlama dilinde Ursina Oyun Motoru kullanılmıştır. Python 3.6 ve üzeri sürümlerde çalışan Ursina, açık kaynak kodlu bir oyun motorudur.

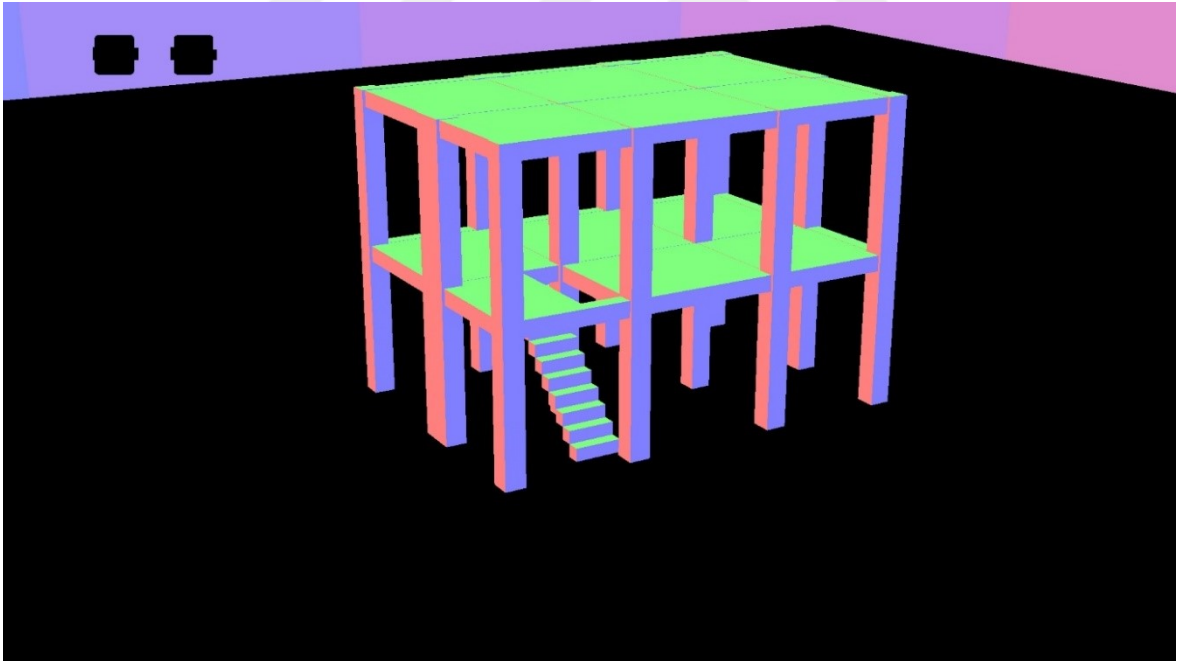


Şekil 3.24: Ursina Koordinat Sistemi

Yapı elemanlarının 3B Modelini oluşturmak için öncelikle ilgili yapının veri tabanından gerekli olan verilerin sorgulama işlemi gerçekleştirilir. Ardından Ursina kütüphanesindeki Entity() sınıfı kullanılarak ilgili elemanın 3B Modeli oluşturulmuş olur. Entity() sınıfı içerisindeki parametreler kullanılarak modelin şekli ile ilgili düzenlemeler yapılır. Örneğin “model” parametresine “plane” değeri atanırsa bu durumda yapının üzerinde durduğu yüzey oluşturulur. Ayrıca “color” parametresi kullanılarak elemanın rengi belirlenmektedir. Renk yerine bir kaplama (doku) kullanılmak istendiğinde bu durumda “texture” parametresi kullanılmaktadır. Örneğin Şekil 3.25’ de Entity() sınıfı içerisindeki “color” parametresi kullanılırken Şekil 4.3’te “texture” parametresi kullanılarak yapıya beton dokusu eklenmiştir. 3B Modeli oluşturulan yapının içerisinde gezinme işlemi için FirstPersonController() sınıfından bir FPS nesnesi oluşturulur. Oluşturulan nesnenin hareket edebilmesi için x, y ve z parametrelerine bir tam sayı değeri girilmelidir. Klavyeden “W, A, S, D, Spacebar” tuşlarını kullanarak oluşturulan nesneyi hareket ettirerek yapı içerisinde gezinilebilir.



Şekil 3.25: A yapısının dokusuz modeli



Şekil 3.26: B yapısının dokusuz 3B Modeli

SQL ile oluşturulan veritabanındaki yapı elemanlarının 3B modellerinin konumlarını belirlemek için koordinat verileri veritabanından çekilir. Ardından çizilen elemanın 3B şeklinin oluşturulabilmesi için en, boy ve yükseklik verileri çekilir. Böylece

3B modelin oluşturulması için veritabanından gerekli tüm bilgiler alınmaktadır. Veri çekme işleminin ardından yapı elemanlarının 3B geometrilerini oluşturabilmek için elemanın en, boy ve yükseklik gibi verileri kullanılmaktadır. Çekilen veriler Entity() sınıfına parametre olarak girilir. Örneğin S01 kolonunun koordinatları Entity() sınıfının “position” parametresine atanır. Kolon elemanın position parametresindeki z değeri “position.z” kat yüksekliğine eşittir. Elemanın en, boy verileri ise “scale” parametresine atanır. Böylece elemanın geometrik şekli oluşturulmuş olur. Örneğin “scale” parametresi “scale=(3, 5, 2)” olan bir elemanın x yönündeki uzunluğu 3 birim, y yönündeki uzunluğu 5 birim ve z yönündeki uzunluğu 2 birim olmaktadır. Fonksiyonel programlama ile Entity() sınıfı kullanılarak her bir elemanın geometrisi oluşturulmaktadır.

Yapının 3B modeli oluşturulduktan sonra kullanıcı tarafından modelin görüntülenmesi için kamera oluşturulmuştur. “EditorCamera()” sınıfı kullanılarak bir kamera nesnesi oluşturulmuştur.

Oluşturulan geometriler ile etkileşime geçebilmek için örneğin yapı içerisinde hareket ederken kolonun, duvarın ya da bir başka yapı elemanının içerisinde geçmemek için elemanların “collider” parametresine “mesh” değişkeni atanmıştır. Mesh değişkeni yapı elemanlarının sınır noktalarını birleştirerek bir çarpışma yüzeyi oluşturmaktadır. Bu çarpışma yüzeyi sayesinde yapı elemanı içerisinde gerçekçi bir şekilde gezinme işlemi yapılmaktadır. Ayrıca oyun motoru içerisinde çarpışma analizi yapabilmek için her elemanın “mesh” değişkenine sahip olması gerekmektedir.

4. VAKA ÇALIŞMALARI

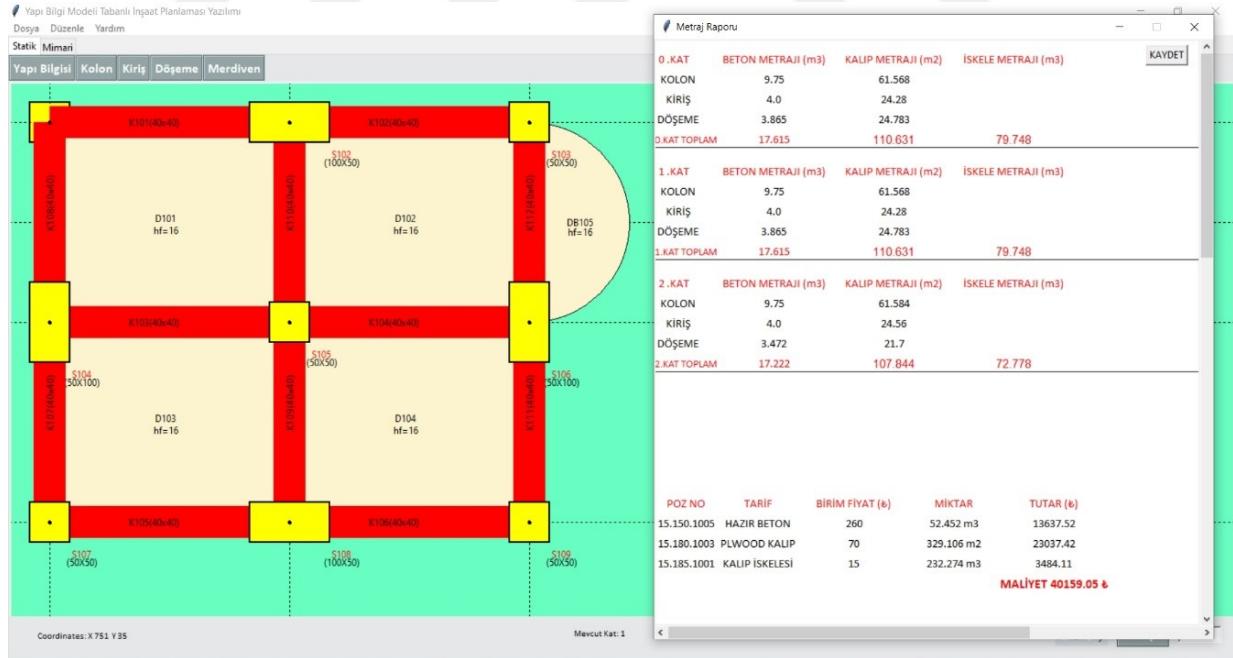
Bölüm 3’te geliştirilme süreci anlatılan yazılımın beton, kalıp, donatı ve iskele metrajlarının hesap yöntemlerini test etmek için vaka çalışmaları yapılmıştır.

4.1 Vaka Çalışması 1

Vaka çalışması 1’de Şekil 4.1 ve Şekil 4.2’deki kalıp planları sunulan yapı elemanlarının beton, kalıp ve iskele metrajları gerçekleştirilmiştir. Şekil 4.1’deki yapı “A yapısı” olarak adlandırılırken Şekil 4.2’deki yapı “B yapısı” olarak adlandırılmaktadır.

A yapısı zemin katla beraber 3 katlı ve her bir katın yüksekliği 3 metre olacak şekilde tasarlanmıştır. Her bir elemana ait butonlar tıklanarak o yapı elemanına ait veriler girilmiştir ve ardından çizimler gerçekleştirilmiştir. B yapısı ise 2 katlı ve her bir katın yüksekliği 3 metre olacak şekilde tasarlanmıştır.

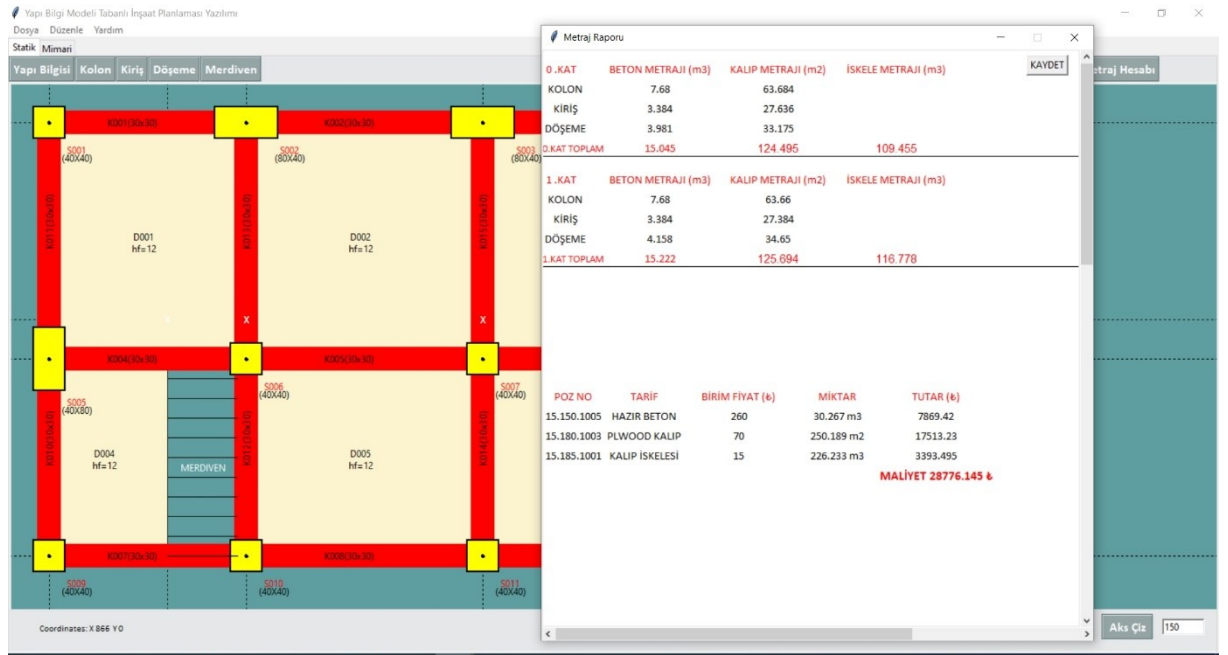
Her bir yapı için ayrı ayrı beton, kalıp ve iskele metrajları hesaplanmış olup sonuçlar Şekil 4.1 ve Şekil 4.2’ de “Metraj Raporu” penceresinde kullanıcıya sunulmaktadır.



Şekil 4.1: A yapısına ait Metraj ve Maliyet Hesabı Raporu

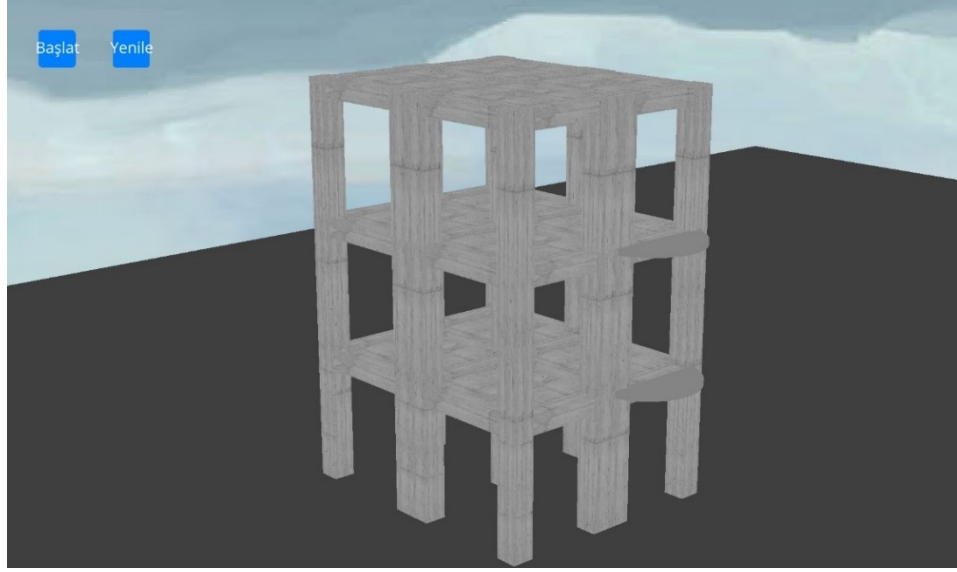
GKA, başarılı bir şekilde test edilmiştir. Kullanıcı ile etkileşim anında bir sorun yaşanmamıştır. Hatasız bir şekilde yazılım çalıştırılarak çizim arayüzü vasıtasıyla

elemanların çizimi gerçekleştirilmiştir. Her bir elemana ait özelliklerin girilmesini sağlayan “açılır pencereler” hatasız bir şekilde açılmıştır. “Metraj Hesabı” butonuna basıldıktan sonra metraj hesaplama işlemleri sırasında da bir hata meydana gelmeden metraj raporu başarılı bir şekilde oluşturulmuştur. Rapor, her bir kat için bilgi sunmaktadır. Metraj dışında elemanların metraj değerlerinin birim fiyat ile çarpılması sonucu bulunan maliyet değeri de raporda yer almaktadır.

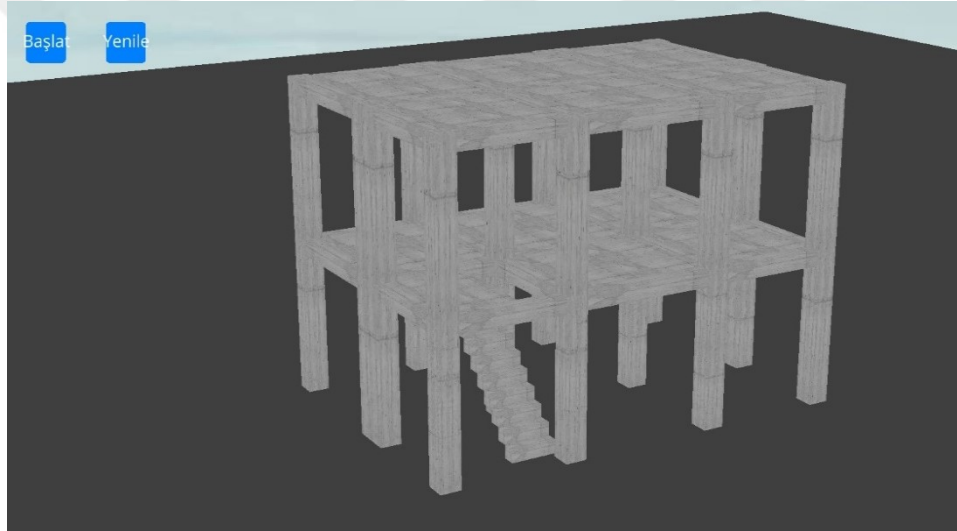


Şekil 4.2: B yapısına ait Metraj ve Maliyet Hesabı Raporu

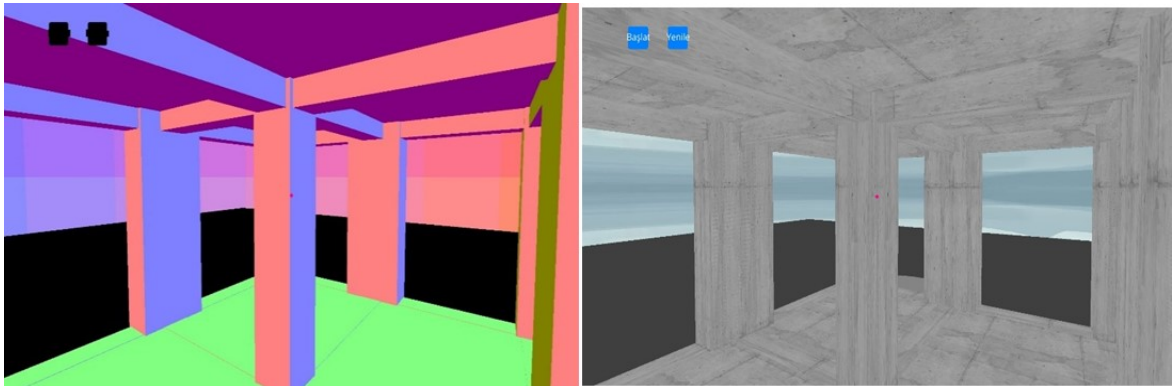
Yapının kalıp planı hazırlandıktan sonra “3B” butonuna basılarak yapının 3B görselleştirilmesi başarılı bir şekilde yerine getirilmiştir. Şekil 4.3 ve Şekil 4.4’ te A ve B yapılarının 3B modelleri sunulmaktadır. Yapıların 3B modelleri “3B” butonuna basılarak otomatik bir şekilde oluşturulup açılır pencerede sunulmuştur. Yapının 3B modeli üzerinde kaydırma, uzaklaştırma ve yakınlaştırma işlemleri başarılı bir şekilde yerine getirilmiştir. Ayrıca yapıların 3B modelinin açıldığı pencerede “Başlat” butonuna basarak Birinci Şahıs Nişancı (BŞN) (*First Person Shooter FPS*) karakteri oluşturulmuş ve yapı içerisinde başarılı bir şekilde gezinme işlemi yerine getirilmiştir. Kullanıcının yapı içerisindeki gezinme işleminin gerçekleştirilmesi Şekil 4.5 ve Şekil 4.6’ da sunulmuştur.



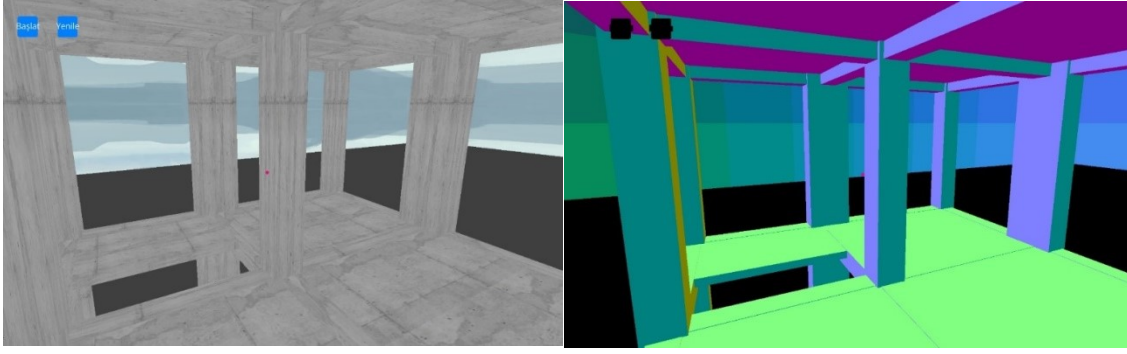
Şekil 4.3: A yapısının beton dokulu 3B Modeli



Şekil 4.4: B yapısının beton dokulu 3B Modeli



Şekil 4.5: A yapısında gezinme işleminin gerçekleştirilmesi

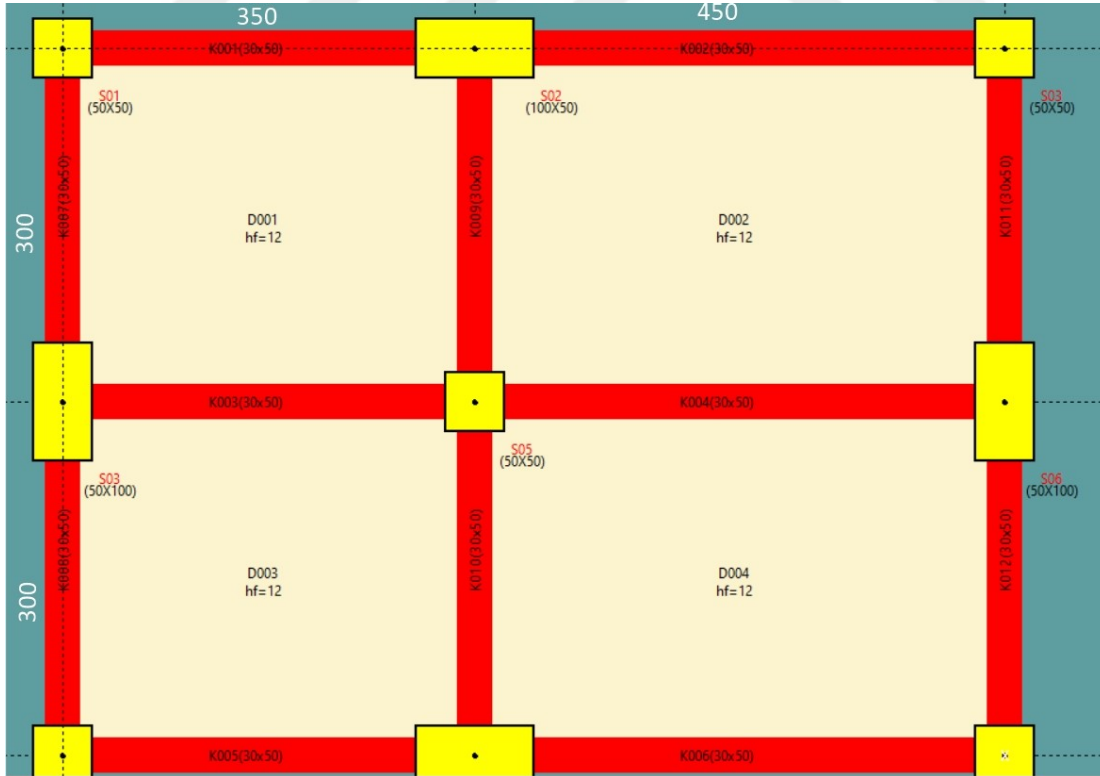


Şekil 4.6: B yapısında gezinme işleminin gerçekleştirilmesi

4.2 Vaka Çalışması 2

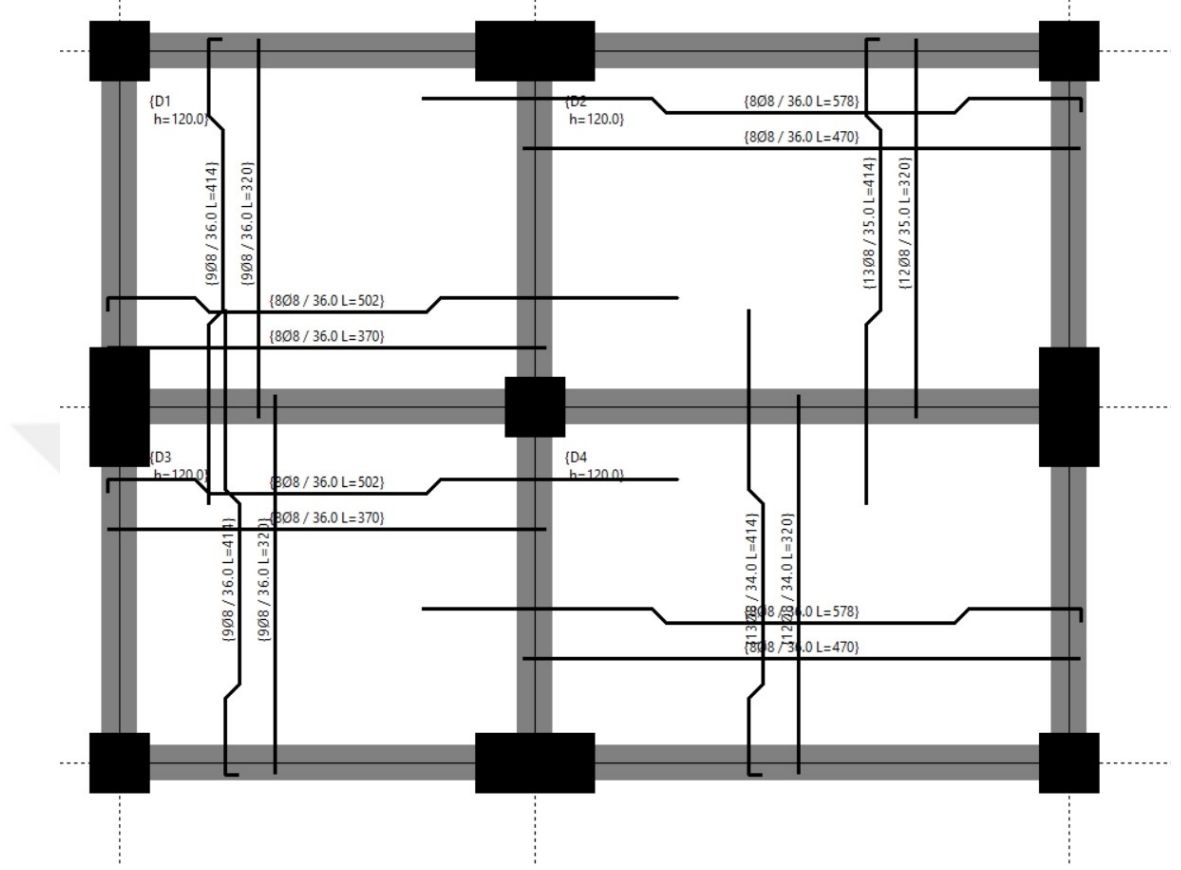
Şekil 4.7’ de sunulan kalıp planındaki betonarme elemanlarının donatı metrajı hesaplanmıştır. Kolon, kiriş ve döşeme için hesaplanan donatı metrajı değerleri sırasıyla Çizelge 4.1, Çizelge 4.2 ve Çizelge 4.3’ te sunulmuştur.

Kolonlar için toplam donatının her kenara eşit yerleştirildiği donatı planına göre donatı yerleşimleri ve meraj hesaplamaları yapılmıştır.



Şekil 4.7: Donatı metrajı hesaplanan yapının Kalıp Planı

Bölüm 3.5.4’te anlatılan yöntemler doğrultusunda donatı metrajları hesaplanmıştır. Döşeme donatı planı otomatik bir şekilde oluşturularak Şekil 4.8’ de sunulmuştur. Metraj, yönetmeliklere uygun bir şekilde ve sınırlandırmaları dikkate alarak hesaplanmıştır.



Şekil 4.8: Döşeme donatı planı

Çizelge 4.1: Kolon donatı metrajı.

AÇIKLAMA	ÇAP (mm)	ADET	UZUNLUK (m)	MİKTAR (kg)
Düşey- S001	20	24	4	236,75
Etriye- S001	8	21	2,04	16,9
Enine Çiroz- S001	8	21	0,63	5,22
Boyuna Çiroz- S001	8	21	0,63	5,22
Düşey- S001	20	28	4	276,21
Etriye- S001	8	21	3,04	25,19
Enine Çiroz- S001	8	63	0,63	15,66
Boyuna Çiroz- S001	8	21	1,13	9,36
Düşey- S003	20	24	4	236,75
Etriye- S003	8	21	2,04	16,9
Enine Çiroz- S003	8	21	0,63	5,22
Boyuna Çiroz- S003	8	21	0,63	5,22
Düşey- S004	22	24	4,1	293,63
Etriye- S004	8	21	3,04	25,19
Enine Çiroz- S004	8	21	1,13	9,36
Boyuna Çiroz- S004	8	63	0,63	15,66
Düşey- S005	20	28	4	276,21
Etriye- S005	8	21	2,04	16,9
Enine Çiroz- S005	8	21	0,63	5,22
Boyuna Çiroz- S005	8	21	0,63	5,22
Düşey- S006	22	24	4,1	293,63
Etriye- S006	8	21	3,04	25,19
Enine Çiroz- S006	8	21	1,13	9,36
Boyuna Çiroz- S006	8	63	0,63	15,66
Düşey- S007	20	24	4	236,75
Etriye- S007	8	21	2,04	16,9
Enine Çiroz- S007	8	21	0,63	5,22
Boyuna Çiroz- S007	8	21	0,63	5,22
Düşey- S008	20	28	4	276,21
Etriye- S008	8	21	3,04	25,19
Enine Çiroz- S008	8	63	0,63	15,66
Boyuna Çiroz- S008	8	21	1,13	9,36
Düşey- S009	20	24	4	236,75
Etriye- S009	8	21	2,04	16,9
Enine Çiroz- S009	8	21	0,63	5,22
Boyuna Çiroz- S009	8	21	0,63	5,22
İnce Donatı 523,020 kg Kalın Donatı 2362, 890 kg Toplam 2885, 910 kg				

Çizelge 4.2: Kiriş donatı metrajı.

AÇIKLAMA	ÇAP (mm)	ADET	UZUNLUK (m)	MİKTAR (kg)
K001-Montaj	12	2	5,40	9,588
K001-Asal	20	4	5,40	53,269
K001-Etriye	8	30	1,60	18,94
K002-Montaj	12	2	6,15	10,92
K002-Asal	20	4	6,15	60,667
K002-Etriye	8	35	1,60	22,097
K003-Montaj	12	2	5,20	9,233
K003-Asal	20	4	5,20	51,296
K003-Etriye	8	31	1,60	19,571
K004-Montaj	12	2	5,95	10,565
K004-Asal	20	4	5,95	58,694
K004-Etriye	8	36	1,60	22,728
K005-Montaj	12	2	5,40	9,588
K005-Asal	20	4	5,40	53,269
K005-Etriye	8	30	1,60	18,94
K006-Montaj	12	2	6,15	10,92
K006-Asal	20	4	6,15	60,667
K006-Etriye	8	35	1,60	22,097
K007-Montaj	12	2	4,50	7,99
K007-Asal	20	4	4,50	44,391
K007-Etriye	8	28	1,60	17,677
K008-Montaj	12	2	4,50	7,99
K008-Asal	20	4	4,50	44,391
K008-Etriye	8	28	1,60	17,677
K009-Montaj	12	2	4,30	7,635
K009-Asal	20	5	4,30	53,022
K009-Etriye	8	29	1,60	18,309
K010-Montaj	12	2	4,30	7,635
K010-Asal	20	5	4,30	53,022
K010-Etriye	8	29	1,60	18,309
K011-Montaj	12	2	4,50	7,99
K011-Asal	20	4	4,50	44,391
K011-Etriye	8	28	1,60	17,677
K012-Montaj	12	2	4,50	7,99
K012-Asal	20	4	4,50	44,391
K012-Etriye	8	28	1,60	17,677

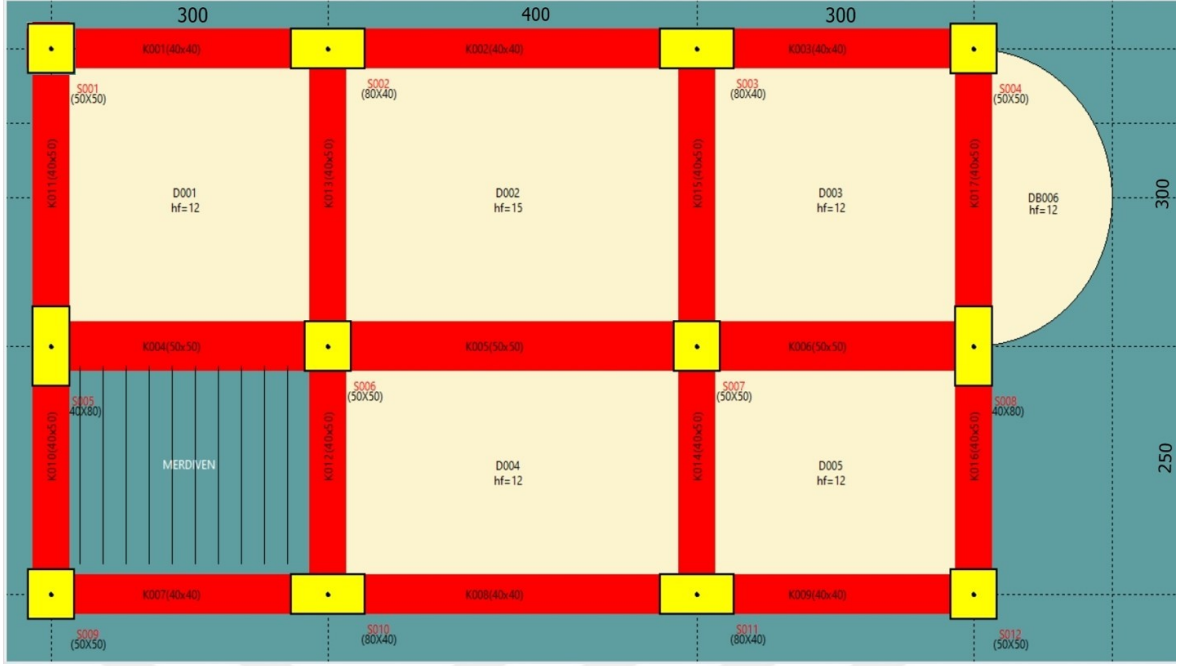
Çizelge 4.3: Döşeme donatı metrajı.

AÇIKLAMA	ÇAP (mm)	ADET	UZUNLUK (m)	MİKTAR (kg)
X pilye - D001	8	8	5,02	15,846
X düz - D001	8	8	3,7	11,68
Y pilye - D001	8	9	4,14	14,702
Y düz - D001	8	9	3,2	11,364
X pilye - D002	8	8	5,78	18,246
X düz - D002	8	8	4,7	14,836
Y pilye - D002	8	13	4,14	21,237
Y düz - D002	8	12	3,2	15,152
X pilye - D003	8	8	5,02	15,846
X düz - D003	8	8	3,7	11,68
Y pilye - D003	8	9	4,14	14,702
Y düz - D003	8	9	3,2	11,364
X pilye - D004	8	8	5,78	18,246
X düz - D004	8	8	4,7	14,836
Y pilye - D004	8	13	4,14	21,237
Y düz - D004	8	12	3,2	15,152

Donatı metraj hesabını gerçekleştiren ve döşeme donatı planını hazırlayan algoritma başarılı bir şekilde test edilmiştir. Şekil 4.8’de görüldüğü gibi donatıların yerleştirilmesinde yönetmelikler tarafından belirtilen donatı kurallarına uyulmuştur. Test işlemi, vaka çalışması kalıp planında gösterildiği üzere karmaşık geometrilere sahip olmayan yapı elemanları üzerinde gerçekleşmiştir. Python programlama dilinin oldukça geniş kitaplıkları ve dinamikliği sayesinde gelecekteki çalışmalarda çok daha karmaşık geometrilere sahip elemanların metraj hesabı için algoritma güncellenebilir ve test edilebilir.

4.3 Vaka Çalışması 3

Vaka çalışması 3, metraj sonuçlarının doğruluğunu test etmek için gerçekleştirilmiştir. Şekil 4.9’ da kalıp planı verilen yapının beton ve kalıp metrajı hesaplanmış olup metraj hesabı yapan yerel 2 farklı yazılım (Y1, Y2) ile karşılaştırılmıştır. Metraj sonuçları Çizelge 4.4’te ve karşılaştırma sonuçları Çizelge 4.5’te verilmiştir.



Şekil 4.9: Vaka çalışması 3 Kalıp Planı (Ergen ve Bettemir, 2022)

Çizelge 4.4: Kalıp ve beton metrajı sonuçları (Ergen ve Bettemir, 2022)

Yer	Metraj Kalemi	Sonuç	Geliştirilen Yazılım	Y1	Y2
Zemin Kat	Beton Metrajı (m ³)	22,9505	22,813	22,95	22,951
	Kalıp Metrajı (m ²)	157,4335	156,811	164,23	157,75
1.Kat	Beton Metrajı (m ³)	23,9308	24,07	23,79	23,939
	Kalıp Metrajı (m ²)	165,7627	165,877	171,23	164,65
Toplam	Beton Metrajı (m ³)	46,8813	46,883	46,74	46,89
Toplam	Kalıp Metrajı (m ²)	323,1962	322,688	335,46	322,40

Çizelge 4.5: Metraj değerlerinin yüzdelik hata değerleri (Ergen ve Bettemir, 2022)

Yüzdelik Hata (%)			
Metraj Kalemi	Geliştirilen Yazılım	Y1	Y2
Beton	%0,004	%-0,301	%0,019
Kalıp	%-0,157	%3,795	%-0,246

5. TARTIŞMA

5.1 Bulguların Değerlendirilmesi

YBM sürecindeki yazılımların kullanılması ile hesaplanan metraj değerlerindeki hatayı azaltmak için birçok çalışma gerçekleştirilmektedir. İnşaat projelerinin yönetilmesinin en önemli aşamalarından birini oluşturan metraj ve yaklaşık maliyet hesaplama aşaması, mühendisler tarafından titizlikle gerçekleştirilen bir aşamadır. Bu çalışmada daha az hata ile yapıların beton, kalıp, donatı ve iskele metrajı yapılmıştır. Geliştirilen yazılım ile ülkemizde kullanılan iki farklı yazılımın beton ve kalıp metrajlarının sonuçları vaka çalışması 3' te karşılaştırılmıştır ve yüzdelik hataları verilmiştir. Hata oranları incelendiğinde beton ve kalıp metrajlarındaki hata oranı diğer programlara kıyasla daha düşüktür. Diğer metraj kalemlerine kıyasla en büyük hata oranları kalıp planı metrajında görülmektedir. Hata oranları incelendiğinde geliştirilen yazılımın kalıp planı metrajındaki başarısı dikkat çekmektedir.

Kalıp metrajının doğru bir şekilde hesaplanması için yapı elemanlarının geometrik özelliklerinin dışında birbirleri ile olan komşuluk ilişkilerinin de bilinmesi gerekmektedir. Komşuluk ilişkileri kullanılarak elemanların birbirleri ile temas ettiği noktaların tespit edilmesi gerekmektedir. Geliştirilen yazılım ile her bir elemanın komşuluk ilişkileri başarılı bir şekilde tespit edilmiştir ve bu başarı kalıp metrajının daha az bir hata ile hesaplanmasında önemli bir rol oynamaktadır. Literatürdeki çalışmalar incelendiğinde komşuluk ilişkileri tespiti için birçok araştırma yapıldığı görülmektedir. Fakat yapılan bu araştırmalar daha çok paralı yazılımlar içerisindeki eklentiler kullanılarak yerine getirilmektedir.

Metraj değerlerinin doğru bir şekilde hesaplanması ile hakediş ödemeleri çok daha doğru bir şekilde yapılacaktır. Kalıp planı metrajındaki yaklaşık %4'lük bir hata milyonlarca TL' lik zarar ile sonuçlanabilir. Kat sayısının artışı ile birlikte metrajdaki hataların daha yüksek değerlerde olması muhtemeldir. Bu durumda zarar katlanarak artacaktır.

Metraj değerlerindeki doğruluğun artmasıyla daha doğru bir yaşam döngü analizi gerçekleştirilecektir. Küresel çaptaki karbon emisyonların çoğunluğunu insan yapımı yapılar oluşturmaktadır. İnşaatla kullanılan malzemelerin (beton, çelik) hatalı metrajı karbon emisyon miktarının hatalı hesaplanmasına sebep olacak ve yaşam döngü analizlerini olumsuz yönde etkileyecektir. Doğru bir analiz yapılmaması sonucu yapının çevre ile olan etkileşimi yanlış bir şekilde değerlendirilmiş olacaktır.

Metraj dışında yapının 3B görselleştirilmesi Ursina oyun motoru kullanılarak başarılı bir şekilde gerçekleştirilmiştir. Yapının 3B modeli projenin farklı aşamalarında proje paydaşlarının daha kısa sürede süreci değerlendirmesi açısından önemlidir. Ayrıca merdiven gibi diğer elemanlara göre modellenmesi nispeten zor olan elemanların modellenmesi başarılı bir şekilde gerçekleştirilerek gelecekteki daha karmaşık yapılarda yapı elemanlarının modellenmesinde referans olacaktır.

Python programlama dili, Sqlite veritabanı motoru ve Ursina oyun motoru arasında verimli bir ilişki kurularak açık kaynaklı programlama dillerinin birbirleri ile etkili bir şekilde kullanılabileceği görülmüştür. Yerel mevzuatlara uygun ve yerel problemlere çözüm üretebilecek bir yazılımın açık kaynaklı programlama dilleri ile geliştirilebileceği görülmektedir.

5.2 Sınırlılıklar

Geliştirilen yazılımdaki metraj hesaplamaları nispeten basit geometrili elemanlara sahip yapılar üzerinde test edilmiştir. Karmaşık geometrilerin metraj hesabı mevcut algoritmanın geliştirilmesi ile yapılabilir. Yazılım geliştirme, uzun süreli ve ekip gerektiren bir iştir. Yüksek lisans kapsamında gerçekleştirilen bu çalışmanın kısıtlı zamanı dikkate alındığında karmaşık geometrilerin metraj hesabı algoritmasını geliştirmek için yeterli vakit bulunamamıştır. Donatı metrajı için gerekli olan veriler statik-betonarme hesabı yapan programlardan alınmıştır. Çünkü statik-betonarme hesabı kendi başına büyük bir hesaplama sürecini oluşturmaktadır. Statik-betonarme hesabı geliştirilen yazılımın hedeflerinden biri olması durumunda ana hedefler olan “daha az hata ile metraj hesabı ve yapının 3B görselleştirilmesi” hedeflerine ulaşmak kısıtlı süre zarfında zor olacaktı. Temel, çatı kısımlarının metrajı yapılmamıştır. Çünkü bahsi geçen bu kısımlar oldukça detaylı süreçlere sahiptir ve herbiri ayrı bir çalışma potansiyeli oluşturmaktadır. Literatür çalışmaları incelendiğinde (Liu vd. (2022), Khataei vd. (2020)) çoğu çalışmada metraj hesaplamalarında temel, çatı ve merdiven gibi elemanların metraj hesaplamaları ile ilgili örneklere dahil edilmediği görülmektedir. Bu eksiklik geliştirilen yazılımın ileride yapılacak çalışmalarda giderilmesi gereken kısımlarını oluşturmaktadır. Sıva, boya ve duvar gibi iş kalemlerinin gelecekteki çalışmalarda metraj hesaplamalarının yapılması ve görselleştirilmesi hedeflenmektedir. Ayrıca ileri çalışmalarda yapının iş programının oluşturulması ve çakışma analizinin gerçekleştirilmesi hedeflenmektedir.

6. SONUÇ VE ÖNERİLER

Daha az maliyet artışı ile işi bitirmek ve malzeme israfını önlemek için metraj hesaplamaları önemlidir. Bu çalışmada bir yapının elemanları arasındaki komşuluk ilişkileri dikkatli bir şekilde incelenip mevcut yazılımlara kıyasla daha az hata ile beton (%0.004 hata oranı) ve kalıp (%0.157 hata oranı) metrajları hesaplanarak daha doğru bir maliyet ve kaynak analizinin yapılması sağlanmıştır. Yapı elemanlarının anlamsal modeli oluşturularak donatı metrajı hesaplanmıştır. Doğru metraj değerleri ile yaşam döngü analizleri daha doğru yerine getirilip yapının çevre ile olan etkileşimi daha sağlıklı değerlendirilecektir. Ayrıca yapının inşaatının ilerleyen aşamalarındaki yanlış metraj hesabı sonucu meydana gelebilecek hataların önüne geçilmiştir.

Yapının 2B çizimini 3B modele dönüştürme yeteneği ve yapı içerisinde gezinebilme özelliği ile proje paydaşlarının projenin farklı süreçlerinde daha hızlı kararlar alması sağlanmıştır. YBM sürecine katkı sağlayacak olan yazılımların açık kaynaklı programlarla geliştirilerek pahalı lisans ücretlerine sahip olan yazılımların kullanımı sınırlandırılmaktadır. Geliştirilen yazılımın gelecekteki çalışmalar ile birlikte daha kapsamlı bir yazılım olması durumunda Türkiye'deki küçük ölçekli inşaat şirketlerinin döviz cinsinden satılan YBM yazılımlarını satın alamaması ve böylece YBM'yi benimseyememesi problemine çözüm olabilecektir.

Python programlama dilinin mühendislik çalışmalarındaki verimliliği çalışma boyunca test edilmiştir. Sonuç olarak süreci hızlandırdığı ve kolaylaştırdığı ortaya çıkmıştır. Mühendislik çalışmalarında diğer dillere göre Python programlama dilinin kullanılması önerilmektedir. Python dışında SQLite ve Ursina oyun motorunun kullanılması çalışmadan daha fazla verim alınmasını sağlamıştır. Kullanılan dillerin yeteneklerinin sınırlandığı alanlarda başka bir dilin destek olarak kullanılması ve aralarında bir ilişki kurulması önerilmektedir.

Gelecekteki çalışmalarda betonarme yapılar dışında çelik yapılar kullanılarak yeni algoritmalar oluşturulabilir. Elde edilen metraj değerleri ile yapının proje yönetimini gerçekleştiren bir yazılım geliştirilebilir. Duvar, sıva ve boya gibi ince işler için metraj hesaplamaları kaba işler için gerçekleştirilen metraj hesaplamalarında kullanılan algoritmanın geliştirilmesi ileride gerçekleştirilecek çalışmalarla sağlanması amaçlanmaktadır.

KAYNAKLAR

- Aranda-Mena, G., Crawford, J., Chevez, A., & Froese, T.** (2009). Building information modelling demystified: does it make business sense to adopt BIM?. *International Journal of managing projects in business*.
- Autodesk Revit.** (t.y.). Autodesk. Retrieved May 4, 2022, from <https://www.autodesk.com.tr/products/revit/overview?term=1YEAR&tab=subscription>
- Azhar, S., Behringer, A., Sattineni, A., & Mqsood, T.** (2012). BIM for facilitating construction safety planning and management at jobsites. In *Proceedings of the CIB-W099 international conference: modelling and building safety, Singapore* (pp. 10-11).
- Azhar, S., Khalfan, M., & Maqsood, T.** (2012). Building information modeling (BIM): now and beyond. *Australasian Journal of Construction Economics and Building, The*, 12(4), 15-28.
- Barlish, K., & Sullivan, K.** (2012). How to measure the benefits of BIM—A case study approach. *Automation in construction*, 24, 149-159.
- Belsky, M., Sacks, R., & Brilakis, I.** (2016). Semantic enrichment for building information modeling. *Computer Aided Civil and Infrastructure Engineering*, 31(4), 261-274.
- BIM Industry Working Group.** (2011). Strategy Paper for the Government Construction Client Group from the BIM Industry Working Group March 2011. *Department of Business, Innovation and Skills. Retrieved February, 19, 2017.*
- Bilgi ve İletişim Teknolojileri.** (t.y.). Wikipedia. Retrieved May 9, 2022, from https://tr.wikipedia.org/wiki/Bilgi_ve_ileti%C5%9Fim_teknolojileri
- Bogdanchikov, A., Zhaparov, M., & Suliyeu, R.** (2013). Python to learn programming. In *Journal of Physics: Conference Series* (Vol. 423, No. 1, p. 012027). IOP Publishing.
- Boon, J., & Prigg, C.** (2012). Evolution of quantity surveying practice in the use of BIM—the New Zealand experience. In *Proceedings of the CIB International Conference on Management and Innovation for a Sustainable Built Environment* (pp. 84-98).
- Bryde, D., Broquetas, M., & Volm, J. M.** (2013). The project benefits of building information modelling (BIM). *International journal of project management*, 31(7), 971-980.
- Bui, N., Merschbrock, C., & Munkvold, B. E.** (2016). A review of Building Information Modelling for construction in developing countries. *Procedia Engineering*, 164, 487-494.
- Bynum, P., Issa, R. R., & Olbina, S.** (2013). Building information modeling in support of sustainable design and construction. *Journal of construction engineering and management*, 139(1), 24-34.
- Choi, J., Kim, H., & Kim, I.** (2015). Open BIM-based quantity take-off system for schematic estimation of building frame in early design stage. *Journal of Computational Design and Engineering*, 2(1), 16-25.

- Dan V. Turda & F.-Zsongor Gobesz** (2005). Python is a C++ made easy for Civil Engineers. *Computational Civil Engineering 2005*”, *International Symposium*
- Du, X.** (2021). Research on engineering project management method based on BIM technology. *Scientific Programming*, 2021.
- Eastman, C. M., Eastman, C., Teicholz, P., Sacks, R., & Liston, K.** (2011). *BIM handbook: A guide to building information modeling for owners, managers, designers, engineers and contractors*. John Wiley & Sons.
- Ergen, F., & Bettemir, Ö. H.** (2022). Development of BIM software with quantity take-off and visualization capabilities. *Journal of Construction Engineering, Management & Innovation*, 5(1), 01–14.
- Franco, J., Mahdi, F., & Abaza, H.** (2015). Using building information modeling (BIM) for estimating and scheduling, adoption barriers. *Universal Journal of Management*, 3(9), 376-384.
- Goucher, D., & Thuraijah, N.** (2012). Advantages and challenges of using BIM: A cost consultant's perspective. In *49th ASC Annual International Conference, California Polytechnic State University (Cal Poly), San Luis Obispo, California*.
- Industry Foundation Classes (IFC)-BuildingSmart.** (t.y.). Building Smart. Retrieved May 15, 2022, from <https://www.buildingsmart.org/standards/bsi-standards/industry-foundation-classes/>
- Ismail, N. A. A. B., Drogemuller, R., Beazley, S., & Owen, R.** (2016). A review of BIM capabilities for quantity surveying practice. In *Proceedings of the 4th International Building Control Conference 2016 (IBCC 2016) [MATEC Web of Conferences, Volume 66]* (pp. 1-7). EDP Sciences.
- İnceoğlu, M. & İnan, B.** (2020). Bilgisayar Destekli Tasarımın Gelişimi: Yeni Bir Mimari Metodoloji Olarak YBM. *GSI Journals Serie C: Advancements in Information Sciences and Technologies*, 3(1), 47-65.
- Jupp, J.** (2017). 4D BIM for environmental planning and management. *Procedia engineering*, 180, 190-201.
- Karagöz, M. E.** (2019). BIM ile yapı yaklaşık maliyeti hesaplama önerisi. *Yapı Bilgi Modelleme*, 1(1), 39-45.
- Khataei, S., Akcamete, A., & Sonmez, R.** (2020). 4D BIM Integrated Productivity Estimation of Construction Projects.
- Khosakitchalert, C., Yabuki, N., & Fukuda, T.** (2020). Automated modification of compound elements for accurate BIM-based quantity takeoff. *Automation in Construction*, 113, 103142.
- Kim, S., Chin, S., & Kwon, S.** (2019). A discrepancy analysis of BIM-based quantity take-off for building interior components. *Journal of Management in Engineering*, 35(3), 05019001.
- Ku, K., Pollalis, S. N., Fischer, M. A., & Shelden, D. R.** (2008). 3D model-based collaboration in design development and construction of complex shaped buildings.
- Liu, H., Cheng, J. C., Gan, V. J., & Zhou, S.** (2022). A knowledge model-based BIM framework for automatic code-compliant quantity take-off. *Automation in Construction*, 133, 104024.

- Millman, K. J., & Aivazis, M.** (2011). Python for scientists and engineers. *Computing in Science & Engineering*, 13(2), 9-12.
- Model View Definition (MVD) - An Introduction.** (y.t.). BuildingSmart. Retrieved May 15, 2022, from <https://technical.buildingsmart.org/standards/ifc/mvd/>
- Nadeem, A., Wong, A. K., & Wong, F. K.** (2015). Bill of quantities with 3D views using building information modeling. *Arabian Journal for Science and Engineering*, 40(9), 2465-2477.
- Omaran, S. M., Alghamdi, A. A., Alharishawi, S. C., & Hains, D. B.** (2019). Integrating BIM and game engine for simulation interactive life cycle analysis visualization. In *Computing in Civil Engineering 2019: Visualization, Information Modeling, and Simulation* (pp. 120-128). Reston, VA: American Society of Civil Engineers.
- Quraishi, A., & Dhapekar, N.** (2021). Applicability of Python in Civil Engineering. *Int. Res. J. Eng. Technol.*, 8(1), 554-556.
- Sabol, L.** (2008). Challenges in cost estimating with Building Information Modeling. *IFMA world workplace*, 1, 1-16.
- Sabol, L.** (2008). Challenges in Cost Estimating with Building Information Modeling. Design and Construction Strategies. LLC, Washington, DC, Available: http://www.dcstrategies.net/files/2_sabol_cost_estimating.pdf (Accessed: March 16th, 2013).
- Sacks, R., Ma, L., Yosef, R., Borrmann, A., Daum, S., & Kattel, U.** (2017). Semantic enrichment for building information modeling: Procedure for compiling inference rules and operators for complex geometry. *Journal of Computing in Civil Engineering*, 31(6), 04017062.
- Sarvade, S. M., & Pore, S. M.** (2019). Use of Python Programming for Interactive Design of Reinforced Concrete Structures. In *National Conference on Exploring New Dimensions in Teaching Learning for Quality Education, Maharashtra, India* (pp. 160-163).
- Stanley, R., & Thurnell, D.** (2014). The benefits of, and barriers to, implementation of 5D BIM for quantity surveying in New Zealand. *Australasian Journal of Construction Economics and Building*, 14(1), 105-117.
- Stine, D. J.** (2012). *Design Integration Using Autodesk Revit 2013*. SDC Publications.
- Succar, B.** (2010). The five components of BIM performance measurement. In *CIB World Congress* (Vol. 14). Salford: United Kingdom.
- Thurairajah, N., & Goucher, D.** (2013). Advantages and challenges of using BIM: A cost consultant's perspective. In *49th ASC Annual International Conference Proceedings* (pp. 1-8). California Polytechnic State University (Cal Poly).
- Topçu, A.** (2019). BETONARME II. *Ders Notu, Eskişehir Osmangazi Üniversitesi*.
- Van Rossum, G.** (2007). Python Programming language. In *USENIX annual technical c Bogdanchikov, A., Zhaparov, M., & Suliye, R. (2013, April). Python to learn programming. In Journal of Physics: Conference Series (Vol. 423, No. 1, p. 012027). IOP Publishing*.
- Wikipedia contributors.** (2006). *ArchiCAD*. Wikipedi. <https://tr.wikipedia.org/wiki/ArchiCAD>

- Woo, J. H.** (2006). BIM (building information modeling) and pedagogical challenges. In *Proceedings of the 43rd ASC national annual conference* (pp. 12-14).
- Yang, B., Zhang, B., Wu, J., Liu, B., & Wang, Z.** (2019). A BIM-based quantity calculation framework for frame-shear wall structure. *Structural Engineering International*, 29(2), 282-291.
- Zhang, J., Chen, Y., Liu, R., & Debs, L.** (2019). Cartesian Points Visualization in Game Simulation for Analyzing Geometric Representations of AEC Objects in IFC. In *Proceedings of the International Symposium on Automation and Robotics in Construction (IAARC)*.
- Zima, K.** (2017). Impact of information included in the BIM on preparation of Bill of Quantities. *Procedia engineering*, 208, 203-210.
- Zuppa, D., Issa, R. R., & Suermann, P. C. (2009).** BIM's impact on the success measures of construction projects. In *Computing in Civil Engineering (2009)* (pp. 503-512).



ÖZGEÇMİŞ

Ad-Soyad : Faruk ERGEN

ÖĞRENİM DURUMU:

- **Lisans** : 2019, İnönü Üniversitesi, Mühendislik Fakültesi, İnşaat Mühendisliği Bölümü

YÜKSEK LİSANS TEZİNDEN TÜRETİLEN ÇALIŞMALAR

- **Ergen, F., & Bettemir, Ö. H.** (2022). Development of BIM software with quantity take-off and visualization capabilities. Journal of Construction Engineering, Management & Innovation, 5(1), 01–14.