

**PARAMETRİK DEĞERLERİ BELİRLENEN
MERMİNİN DEFORME OLABİLEN YÜZEYLERE
VERDİĞİ ZARARIN MODELLENMESİ VE
BİLGİSAYAR GRAFİKLERİ İLE SİMÜLASYONU**

**2014
YÜKSEK LİSANS TEZİ
BİLGİSAYAR MÜHENDİSLİĞİ**

Rafet DURGUT

**PARAMETRİK DEĞERLERİ BELİRLENEN MERMİNİN DEFORME
OLABİLEN YÜZEYLERE VERDİĞİ ZARARIN MODELLENMESİ VE
BİLGİSAYAR GRAFİKLERİ İLE SİMÜLASYONU**

Rafet DURGUT

**Karabük Üniversitesi
Fen Bilimleri Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalında
Yüksek Lisans Tezi
Olarak Hazırlanmıştır**

**KARABÜK
Haziran 2014**

Rafet DURGUT tarafından hazırlanan “PARAMETRİK DEĞERLERİ BELİRLENEN MERMİNİN DEFORME OLABİLEN YÜZEYLERE VERDİĞİ ZARARIN MODELLENMESİ VE BİLGİSAYAR GRAFİKLERİ İLE SİMÜLASYONU” başlıklı bu tezin Yüksek Lisans Tezi olarak uygun olduğunu onaylarım.

Yrd. Doç. Dr. İsmail KURNAZ

.....

Tez Danışmanı, Bilgisayar Mühendisliği Anabilim Dalı

Bu çalışma, jürimiz tarafından oy birliği/oy çokluğu Bilgisayar Mühendisliği Anabilim Dalında Yüksek Lisans tezi olarak kabul edilmiştir. 27/06/2014

Ünvanı, Adı SOYADI (Kurumu)

İmzası

Başkan : Prof. Dr. Mehmet AKBABA (KBÜ)

.....

Üye : Yrd. Doç. Dr. Ahmet Hanifi ERTAŞ (KBÜ)

.....

Üye : Yrd. Doç. Dr. İsmail KURNAZ (KBÜ)

.....

...../...../2014

KBÜ Fen Bilimleri Enstitüsü Yönetim Kurulu, bu tez ile, Yüksek Lisans derecesini onamıştır.

Prof. Dr. Mustafa BOZ

.....

Fen Bilimleri Enstitüsü Müdürü

“Bu tezdeki tüm bilgilerin akademik kurallara ve etik ilkelere uygun olarak elde edildiğini ve sunulduğunu; ayrıca bu kuralların ve ilkelerin gerektirdiği şekilde, bu çalışmadan kaynaklanmayan bütün atıfları yaptığımı beyan ederim.”

Rafet DURGUT

ÖZET

Yüksek Lisans Tezi

PARAMETRİK DEĞERLERİ BELİRLENEN MERMİNİN DEFORME OLABİLEN YÜZEYLERE VERDİĞİ ZARARIN MODELLENMESİ VE BİLGİSAYAR GRAFİKLERİ İLE SİMÜLASYONU

Rafet DURGUT

Karabük Üniversitesi

Fen Bilimleri Enstitüsü

Bilgisayar Mühendisliği Anabilim Dalı

Tez Danışmanı:

Yrd. Doç. Dr. İsmail KURNAZ

Haziran 2014, 56 sayfa

Bu çalışmada; bilgisayar grafikleri kullanılarak, parametrik değerleri bilinen mermilerin deforme olabilen yüzeylere verdiği zararın bilgisayar grafikleri ile modellenmesi ve benzetimi gerçekleştirilmiştir. Hafif silah mühimmatı türündeki mermi için kullanılan parametrik değerler; ilk çıkış hızı, kütlesi, sürtünme katsayısı ve boyutudur. Bu çalışmada, parametrik değerleri birbirlerinden farklı olan 5 mermi çeşidinin atış sonrası parametrik özellikleri bilinen 5 farklı malzeme yüzeyi üzerinde oluşturdukları hasarlar gösterilmiştir. Mermi türü ve yüzey çeşidi sayısı parametrik değerleri girilmesi koşulu ile arttırılabilmektedir. Uygulamada merminin, yüzeye vereceği hasar, hem sayısal hem de görsel olarak gösterilmektedir. Uygulamada yer alan 3B öğeler Blender ile modellenip, anime edilmiştir. Malzemelerin parametrik değerleri ANSYS yazılımından alınmıştır. Benzetim işleminin gösterildiği uygulama OGRE3D kütüphanesi kullanılarak C++ programlama dili ile hazırlanmıştır.

Anahtar Sözcükler : Bilgisayar grafikleri, 3B modelleme, doku eşleme, 3B simülasyon, görüntü gerçekleştirme.

Bilim Kodu : 902.1.104

ABSTRACT

M. Sc. Thesis

MODELLING THE PENETRATION OF A BULLET WHICH PARAMETRIC VALUES SPECIFIED TO THE DEFORMABLE SURFACES AND ITS SIMULATION BY COMPUTER GRAPHICS

Rafet DURGUT

Karabük University

Graduate School of Natural and Applied Sciences

Department of Computer Engineering

Thesis Advisor:

Assist. Prof. Dr. İsmail KURNAZ

June 2014, 56 pages

In this study the effect to the deformable surfaces of bullets which parametric values already known has been modeled and simulated by using computer graphics. The parametric values of light gun ammunition can be enumerated such as initial velocity, mass, coefficient of friction, and size. In the study five kinds of bullet and five kinds of material surfaces have been selected. The types of bullet and surface can be incremented by applying their parametric values. The effect of bullet to the surface is shown both numerically and visually in the application. The 3d objects in the application have been modeled and animated with Blender 2.67. The parametric values of materials have been taken from ANSYS. The application that shows simulation process is developed by an API called as OGRE3D using C++ programming language.

Key Word : Computer graphics, 3D modelling, texture mapping 3D simulation, rendering.

Science Code : 902.1.104

TEŐEKKÜR

Bu tez alıőmasının planlanmasında, araőtırılmasında, yürütülmesinde ve oluşumunda ilgi ve desteęini esirgemeyen, engin bilgi ve tecrübelerinden yararlandığım, yönlendirme ve bilgilendirmeleriyle alıőmamı bilimsel temeller ışığında őekillendiren sayın hocam Yrd. Do. Dr. İsmail KURNAZ'a sonsuz teőekkürlerimi sunarım.

Ayrıca alıőmamın mekanik kısmında bana yol gösteren ve yardımcı olan sayın hocam Yrd. Do. Dr. Ahmet Hanifi ERTAŐ'a sonsuz teőekkürlerimi sunarım.

Sevgili aileme manevi hiçbir yardımı esirgmeden yanımda oldukları için tüm kalbimle teőekkür ederim.

İÇİNDEKİLER

	<u>Sayfa</u>
KABUL.....	ii
ÖZET.....	iv
ABSTRACT.....	vi
TEŞEKKÜR.....	viii
İÇİNDEKİLER	ix
ŞEKİLLER DİZİNİ.....	xii
ÇİZELGELER DİZİNİ	xiii
SİMGELER VE KISALTMALAR DİZİNİ	xiv
BÖLÜM 1	1
GİRİŞ	1
BÖLÜM 2	4
BİLGİSAYAR GRAFİKLERİ.....	4
2.1. MODELLEME.....	5
2.1.1. Blender.....	6
2.1.2. Autodesk Maya.....	6
2.1.3. Autodesk 3DS Max	6
2.2. GÖRSEL GE RÇEKLEME.....	7
2.2.1. OpenGL	8
2.2.2. DirectX	8
2.2.3. Ogre3D	9
2.2.4. OpenSceneGraph	9
2.3. BENZETİM.....	10
2.3.1. Bullet Physics	11
2.3.2. Newton Dynamics Engine (NDE)	11
2.3.3. Open Dynamics Engine (ODE)	11
2.3.4. Physx	12

	<u>Sayfa</u>
BÖLÜM 3	13
BALİSTİK VE KATI CİSİM MEKANİĞİ	13
3.1. BALİSTİK.....	13
3.1.1. İç Balistik.....	13
3.1.2. Dış Balistik	14
3.1.3. Terminal Balistik	14
3.2. KATI CİSİM MEKANİĞİ	15
3.2.1. Mermi Hareketi.....	16
3.2.2. Ötelenme ve Dönme	16
3.2.3. Kuvvet ve İvme.....	17
3.2.4. İş ve Enerji.....	18
3.2.5. İtme ve Momentum	19
3.2.6. Malzeme Modeli	19
BÖLÜM 4	21
ÖNCEKİ ÇALIŞMALAR.....	21
BÖLÜM 5	25
MERMİ HASAR SİMÜLASYONU	25
5.1. MODELLERİN HAZIRLANMASI.....	25
5.1.1. Mermi Modelleme	27
5.1.2. Karakter Modelleme	31
5.2. MODELLERİN GÖRSEL GERÇEKLEME İÇİN HAZIRLANMASI.....	33
5.3. GÖRSEL GERÇEKLEME.....	34
5.4. SİMÜLASYON.....	38
5.4.1. Fiziksel İşlemler.....	38
5.4.2. Grafikselsel İşlemler	43
BÖLÜM 6	46
BULGULAR.....	46

	<u>Sayfa</u>
BÖLÜM 7	50
SONUÇLAR	50
KAYNAKLAR	52
ÖZGEÇMİŞ	56

ŞEKİLLER DİZİNİ

	<u>Sayfa</u>
Şekil 2.1. OpenGL temel öğeleri.....	8
Şekil 3.1. Mermi hareket eğrisi	16
Şekil 3.2. Ötelenme hareketi	17
Şekil 5.1. 7.62 x 51 NATO mermi ölçüleri.....	27
Şekil 5.2. Blender açılış ekranı.....	27
Şekil 5.3. Blender modları seçim ekranı	28
Şekil 5.4. Düzenleme kipi yüzey seçimi	29
Şekil 5.5. Çıkartım işlemi uygulanmış küp modeli	29
Şekil 5.6. Ölçeklendirme işlemi uygulanmış model	30
Şekil 5.7. 7.62 x 51 NATO mermi üç boyutlu görünümü.....	31
Şekil 5.8. Karakterin 3 boyutlu ve kemik sistemli modellenmiş hali	32
Şekil 5.9. Animasyon ekranı	32
Şekil 5.10. OgreXMLConverter işlem ekranı	34
Şekil 5.11. Ogre3D sahne grafi	35
Şekil 5.12. Varlıkların sahne üzerinde görünümü.....	35
Şekil 5.13. Ogre3D ile merminin görsel gerçekleştirilmesi	37
Şekil 5.14. Voksel yapısı.....	41
Şekil 5.15. Çarpışma kontrolü şeması.....	42
Şekil 5.16. Malzeme ve mermi seçim menüsü.....	44
Şekil 5.17. Hasar almış malzeme yüzeyi	45
Şekil 6.1. Aynı uzaklıktan ve aynı atış derecesiyle atılan mermilerin alüminyum malzeme yüzeyine uyguladıkları itme kuvvetleri.	46
Şekil 6.2. Aynı uzaklıktan aynı atış derecesiyle atılan 7.62 x 51 mermisinin farklı malzeme yüzeylerine uyguladığı itme kuvvetleri.	47
Şekil 6.3. Aynı uzaklıktan atılan mermilerin hedefe ulaşıncaya kadar olan bir eksenindeki hızları.....	48
Şekil 6.4. Aynı uzaklıktan atılan mermilerin hedefe ulaşıncaya kadar olan y eksenindeki hızları..	49

ÇİZELGELER DİZİNİ

	<u>Sayfa</u>
Çizelge 4.1. Kullanılan mermi tipleri ve parametreleri.....	26
Çizelge 4.2. Kullanılan malzeme yüzeyleri ve parametreleri	26

SİMGELER VE KISALTMALAR DİZİNİ

SİMGELER

- Ag : gümüş
Al : alüminyum
Pb : kurşun
Fe : demir
v : hız
p : momentum
F : kuvvet
t : zaman
m : kütle
E_k : kinetik enerji
E_p : potansiyel enerji
E_{ısı} : ısı enerjisi
U : iş
g : yer çekimi ivmesi

KISALTMALAR

2B	: 2 Boyutlu
3B	: 3 Boyutlu
SEM	: Sonlu Elemanlar Metodu
BDT/BDÜ	: Bilgisayar Destekli Tasarım / Bilgisayar Destekli Üretim
API	: Application Programming Interface (Uygulama Programlama Arayüzü)
MIT	: Massachusetts Institute of Technology (Massachusetts Teknoloji Enstitüsü)
ODE	: Open Dynamics Engine
NDE	: Newton Dynamics Engine
SI	: Système international d'unités (Uluslar arası Birim Sistemi)
MKEK	: Makina ve Kimya Endüstrisi Kurumu
XML	: Extensible Markup Language (Genişletilebilir işaretleme dili)
AISI 4340	: Çelik 4340
Ogre3D	: Object-Oriented Graphics Render Engine (Nesneye Dayalı Grafik Gerçekleme Motoru)
MHS	: Mermi Hasar Simülasyonu

BÖLÜM 1

GİRİŞ

Mikroişlemci üretiminin lider firmalarından Intel şirketinin kurucusu Gordon Moore kendi ismiyle anılan Moore yasası ile her 18 ayda bir, öncekinin iki katı transistör kullanan bir işlemci satın alınabileceğini savunan bir öngöründe bulunmuştur. Bu yasa günümüzde bilgisayar teknolojisinin hızlı gelişiminin temel unsuru olarak kabul edilir [1]. Bilgisayar mimarisinde meydana gelen bu hızlı gelişim, bu teknolojiyi kullanan alanların da gelişmesini sağlamıştır. Bu alanlardan biri de bilgisayar grafikleridir. Bilgisayar grafiklerindeki gelişimin temel sebebi olarak Moore yasasının yanı sıra grafik işlemcileri teknolojisindeki hızlı gelişim de gösterilebilir. Kullanılan donanım özelliklerinin geliştirilmesi, kullanılacak olan bilgisayar grafiklerinin de kalitesini belirleyen etkenlerden birisidir [2].

Son yıllarda bilgisayar grafikleri, birçok disiplin ile beraber kullanılmaktadır. Tıp, mühendislik, savunma sanayi gibi disiplinlerde bilgisayar grafikleri ile yapılan uygulamaların kullanılma oranlarında artış görülmektedir [3]. Örneğin, Üç Boyutlu Etkileşimli Sanal Anatomi Laboratuvarı Tasarısı ve Uygulaması ile Dokuz Eylül Üniversitesi tıp fakültesinde gerçek zamanlı anatomik modellerin 3B olarak incelenmesi [4] için bir benzetim uygulaması geliştirilmiştir. Günümüzde benzetim uygulamaları getirdiği yararlar ile, birçok disiplinin olmazsa olmazı haline gelmektedir.

Bilgisayar grafiklerinin çalışma alanları modelleme, gerçekleştirme ve benzetim olarak sınıflandırılabilir [2]. Bu sınıflandırma gruplarında her grup, sonraki grup için ön koşul niteliğinde tanımlanabilir. Örneğin gerçekleştirme yapılabilmesi için modelleme, benzetim yapılabilmesi için gerçekleştirme yapılması gereklidir, denilebilir. Fakat modelleme, gerçekleştirme ve benzetim alt dalları birbirlerinin devamı niteliğinde olmakla birlikte, birbirlerinden farklı işlemler içermektedirler.

Mühendisliđi birçok dalında bilgisayar grafiklerinden yararlanılmaktadır. Özellikle yüksek maliyet gerektiren işlerde projelerin önce bilgisayarlarda simüle edilmesi hem zaman, hem de maliyet bakımından oldukça tasarruf sağlayabilmektedir. Birden fazla seçenek olduđu durumlarda en iyi durumun hangisi olduđunu bulmak için, tüm durumların sonuçlarının incelenmesi gerekmektedir. Gerçek hayatta bu işi gerçekleştirmek oldukça fazla zaman ve maliyet gerektirmektedir. Örneđin BDT/BDÜ yazılımları ile tasarım, analiz ve imalat gibi süreçlerde karşılaşılabilecek durumlar bilgisayar grafikleri kullanılarak simüle edilebilmektedir. Böylece zaman ve maliyet tasarrufu sağlanabilir.

Hazırlanan bu çalışmada, parametrik özellikleri farklı olan mermilerin atışı ile bu atış sonucunda malzeme yüzeylerine verdikleri hasar modellenmiş ve simüle edilmiştir. Bu çalışma yedi bölümden oluşmaktadır. Çalışmanın ilk bölümünde; bilgisayar grafikleri ile diđer bilim dalları arasındaki ilişkiler hakkında kısaca bilgi verilmiştir.

Çalışmanın ikinci bölümünde; bilgisayar grafiklerinin genel tanımı, kullanım alanları ve alt dalları belirtilmiş ve çalışmada kullanılan Blender, Ogre3D ve Bullet Pyhsics isimli kütüphane ve uygulamaların özellikleri verilmiştir.

Çalışmanın üçüncü bölümünde; balistik bilimi ve çalışmanın teorik dayanađı olan katı cisim mekaniđi incelenmiştir. Bu itibar ile balistik biliminin alt bilim dalları olan iç, dış ve terminal balistik konuları ayrıntılı olarak ele alınmış, sonrasında katı cisim mekaniđinin temel uygulamaları hakkında bilgi verilmiştir.

Çalışmanın dördüncü bölümünde; daha önce bilgisayar grafikleri ve mermi hasarı alanında yapılmış olan çalışmalar hakkında bilgi verilmiştir.

Çalışmanın beşinci bölümünde; geliştirilen uygulama hakkında bilgi verilmiştir. Uygulamada kullanılan model ve animasyonların hazırlanma aşamaları, katı cisim dinamiklerinin uygulanması, benzetim işlemleri ayrıntılı olarak anlatılmıştır. Ayrıca uygulama geliştirilirken karşılaşılan sorunlar ve bu sorunlar için üretilen çözüm önerileri de bu bölümde ele alınmıştır.

Çalışmanın altıncı bölümünde çalışma sonucunda elde edilen veriler grafikler halinde sunulmuş ve bu grafikler hakkında yorumlar yapılmıştır.

Çalışmanın yedinci ve son bölümünde bu tez çalışmasında elde edilen sonuçlar belirtilmiştir. Çalışmanın geleceği için önerilerde bulunulmuştur.

BÖLÜM 2

BİLGİSAYAR GRAFİKLERİ

Bilgisayar grafiđi, bilgisayar kullanılarak 2B/3B bir modelin görüntüsünü elde etme, görüntüyü ekranda gösterme, görüntü üzerinde çalışma gibi işlemlerin yapıldığı çalışma alanıdır. Her ne kadar bilgisayar grafikleri denildiğinde akıllara sadece görsel olarak bir nesnenin gösterilmesi gelse de, işlevleri sadece bununla sınırlı değildir. Bilgisayar grafikleri veri görselleştirme (data visualization), sanal gerçeklik (virtual reality), görüntü işleme (image processing), bilgisayar destekli tasarım (computer aided design) gibi birçok alanda etkili olarak kullanılmaktadır.

Teknolojinin çok hızlı gelişmesi, bilgisayar grafiklerinin kullanımını da etkilemiştir. 2B ekran üzerinde, 2B görüntüler yerine daha karmaşık olan ve daha yüksek kaliteye sahip 3B görüntüler kullanılmaya başlanmıştır. Bu gelişmeyle beraber, bilgisayar grafiklerinin özellikle eğlence ve oyun sektörlerindeki kullanımında, büyük bir oranda artış yaşanmış ve neticede örneğın bir film sadece bilgisayar grafikleri ile yapılır hale gelmiştir. Son dönemlerde yapılan filmlerin kalitesini belirleyen parametrelerden birisi de; bilgisayar grafiklerinin kullanılma oranı olmuştur [3].

Bilgisayar grafiklerini alt kategorilere ayırmak zordur. Fakat çoğunluk tarafından kullanılan temel kategoriler vardır. Bunlar, modelleme (modelling), gerçekleştirme (rendering) ve animasyon (animation) olarak sıralanabilir. Bu üç temel kategorinin yanında kullanıcı etkileşimi, sanal gerçeklik, görselleştirme, görüntü işleme ve 3B tarama gibi alt kategoriler de oluşturulabilir. Bu alanların en çok kullanıldığı uygulamalar ise aşağıdaki gibi sıralanabilir [5];

- Bilgisayar Oyunları
- Animasyon Filmleri
- Özel Efektli Filmler

- BDT/BDÜ Yazılımları
- Benzetim
- Tıbbi Görüntüleme
- Veri Görselleştirme

2.1. MODELLEME

Modelleme, bir sistem ya da sistem parçasının davranışının gerçek hayattaki davranışına, matematiksel ve fiziksel değerlerine uygun olarak oluşturulması işlemidir. Modelleme yapılırken temel esas, modeli oluşturulacak olan nesnenin sahip olduğu özelliklerin gerçeği yansıtmasıdır. Gerçekçi olmayan bir model kullanılması ise kullanıcılarda yanlış ya da istenmeyen algılara sebep olabilir.

Bilgisayar grafikleri için kullanılan modelleme, bir sistem ya da sistem parçasının görüntüsünün gerçek hayattaki görüntüsüne, matematiksel ve fiziksel esaslara bağlı kalmak koşulu ile bilgisayar ortamında gerçekleştirme işlemidir.

Bilgisayar grafikleri ile modelleme için kullanılacak birçok yol ve yöntem vardır. En çok kullanılan yöntemler arasında eğriler, yüzeyler ve poligonlar bulunmaktadır. Bu yöntemler arasında en sık kullanılanı poligon modellemedir. Poligon modellemede, nesnenin iç kısmını dış dünyadan ayıran yüzey geometrik olarak tanımlanır. Eğri ve yüzey modelleme ise, nesnenin uzayda kapladığı alan göz önünde bulundurularak modelleme yapılır. Yapılacak olan modellemenin amacı ve gereksinimlerine göre kullanılacak olan yöntem belirlenmelidir [6].

Bilgisayar grafikleri alanının büyümesi ve kişisel bilgisayarların da bu alanda kullanılabilmesiyle beraber daha hızlı ve daha kolay modelleme yapılabilmesi için uygulamalara ihtiyaç duyulmuştur. Bu ihtiyaçları gidermek için Blender, Autodesk Maya, Autodesk 3DS Max gibi birçok modelleme programı geliştirilmiştir. Günümüzde kullanıcılar kendi istekleri doğrultusunda bu modelleme programları arasından tercih yapmaktadırlar.

2.1.1. Blender

Blender, 1993 yılında geliştirilmeye başlanmıştır. Kararlı hale ise 1994 yılında geçmiştir. Görsel ara yüzü sayesinde 2B ve 3B içerik oluşturulmasını sağlayan bir yazılımdır. Açık kaynak kodlu olan Blender, platformdan bağımsız olarak çalışmaktadır. Kendi içerisinde “Bullet Physics” isimli fizik motoruna sahiptir. Dünya çapında 100 gönüllü tarafından geliştirilmeye devam etmektedir. Blender’in güncel sürümü 2.69’dur. Blender, kayıtlı 150 bin kullanıcı tarafından kullanılmaktadır. Eklenti yazımı için kullanılan dil Python’dur [7].

Blender’in temel işlevleri; modelleme, doku eşleme, ışıklandırma, animasyon ve video çekim sonrası düzenleme şeklinde sıralanabilmektedir. Ayrıca gelişmiş kemik sistemi kullanılarak daha gerçekçi animasyonlar hazırlanabilmektedir [8].

2.1.2. Autodesk Maya

Maya olarak da bilinen Autodesk Maya, 3B bilgisayar grafikleri yazılımıdır. Özellikle film sektöründe, görsel efektlerin oluşturulmasında kullanılır. Windows, Mac OS ve Linux işletim sistemlerinde çalışabilmektedir. Alias Systems tarafından 1998 yılında geliştirilmeye başlanan yazılımı, 2005 yılında Autodesk firması satın almıştır. Bu tarihten itibaren Autodesk Maya olarak bilinmektedir [9].

Maya, Bullet Physics isimli fizik motorunu kullanmaktadır. Maya, içerisinde kullanılmaya hazır birçok fiziksel işlemi barındırmaktadır. Maya, bu tür fiziksel işlemleri gerçekleştirmeye olanak sağlayan akışkan benzetimi, giysi benzetimi, saç benzetimi, parçacık benzetimi gibi birçok eklentiye sahiptir. Harici olarak eklenti yazmak ve Maya ile entegre etmek de mümkündür. Bu tür işlemler için Maya Embedded Language olarak tanımlanan özel bir programlama dili kullanılır [9].

2.1.3. Autodesk 3DS Max

3DS Max, Autodesk firması tarafından geliştirilen 3B modelleme ve animasyon programıdır. Temel işlevleri, modelleme, animasyon, görüntüleme, birleştirme, parçacık sistemi ve karakter modelleme olarak sıralanabilir. Genel kullanım alanları

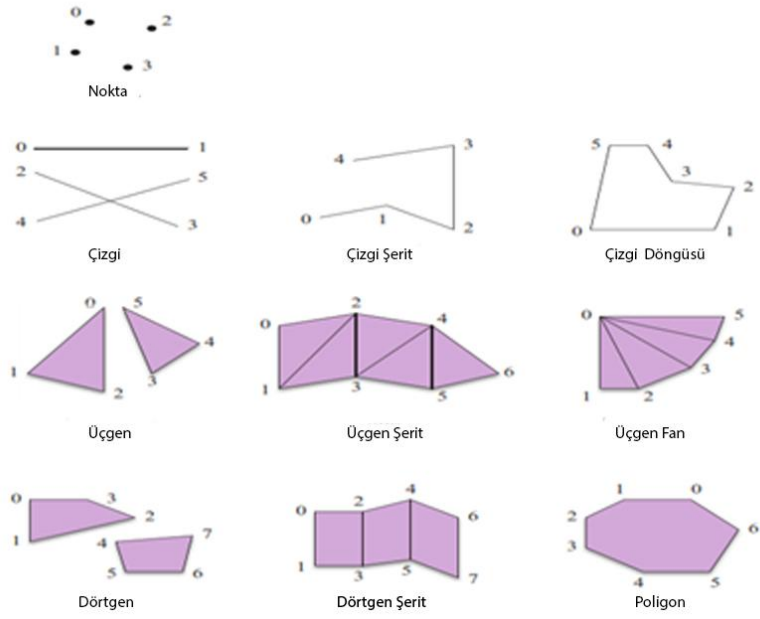
ise filmler için özel efekt oluşturma, mimari sunumlar ve endüstriyel ürün tasarımları olup, yaygın olarak da bilgisayar grafiklerinin temel işlevlerinin gerçekleştirilmesinde kullanılmaktadır. Eklenti yazımı için gerekli olan programlama dili Python'dur [10].

3DS Max mimari tasarımlarda daha önde olan bir 3B modelleme yazılımıdır. Kendi içerisinde hazır bulunan görsel materyal kütüphanesi vardır. Bu materyallerin ölçüleri değiştirilebilir ve yeni modeller oluşturulabilir. Fizik kütüphanesi olarak "Havok" firmasının ürettiği "Reactor" isimli kütüphaneyi kullanmaktadır. 3DS Max ücretli bir yazılım olup, sadece 30 günlük deneme sürümü ücretsiz olarak kullanılabilir [10].

2.2. GÖRSEL GERÇEKLEME

Görsel gerçekleştirme, bilgisayar ekranında görüntülenecek olan tüm nesnelerin nasıl görüntüleneceğinin bilgisayar tarafından hesaplanması ve bu hesaplama sonucunda nesnelerin ekranda gösterilmesi işlemidir. Bu işlem sahne olarak adlandırılan, nesnelerin konum ve pozisyonlarının belirli olduğu ana nesne üzerinde gerçekleşir. Sahne dışında olan nesneler ekranda gösterilmez. Her bir nesnenin, her bir noktası için hesaplama yapılacağı için bilgisayarın yüksek performanslı olması arzu edilir [11].

Bilgisayar grafikleri ile gerçekleştirilen bir sahnede, birbirinden ayrı nesneler bulunabilir. Bu nesnelerin geometrik formları, köşe noktalarının bulunduğu bir dizi ve köşe noktalarının birbirleriyle bağlanırken hangi grafiksel bağlantı şeklini kullanacağını belirtir. Grafiksel bağlantı şeklinin (graphics primitive) farklı türleri mevcuttur. Örnek olarak; OpenGL grafik kütüphanesinin kullandığı 10 temel bağlantı Şekil 2.1'de görüntülenmektedir. Grafik donanımları, noktalar dizisinden, çizgilerin birleşiminden ya da doldurulmuş çokgenlerden gerçekleştirme yapabilme yeteneğine sahiptirler. 3 Boyutlu bir model genellikle; her köşenin noktası belirlenen üçgen listesi ile ifade edilir [12].



Şekil 2.1. OpenGL temel öğeleri [11].

Grafik işlemleri gerçekleştirmek için bir çok kütüphane bulunmaktadır. OpenGL, DirectX, Ogre3D, Delta3D, OpenSceneGraph gibi kütüphaneler yaygın olarak kullanılmaktadır.

2.2.1. OpenGL

Grafik kütüphaneleri arasında en yaygın kullanılanı OpenGL'dir. OpenGL Silicon Graphics Inc. tarafından hazırlanmış, açık kaynak kodlu ve ücretsiz dağıtılan grafik kütüphanesidir. Bilgisayarların grafik donanımı için gelişmiş destek sunmaktadır. 2B ve 3B bilgisayar grafiklerini ekranda göstermek için kullanılan bir yazılım arayüzüdür (API). Bu arayüz sayesinde arayüzü kullanan yazılımcı üç boyutlu, yüksek kaliteli grafikler üretebilir. Windows, Linux, Mac OS gibi birçok işletim sisteminde ve Playstation3 gibi oyun konsolunda desteklenen bu kütüphane, deneysel, bilimsel amaçlar için yaygın olarak kullanılmaktadır [13].

2.2.2. DirectX

Microsoft tarafından geliştirilen DirectX, başta Microsoft olmak üzere video oyunları ve çoklu ortam yazılımlarının rahat, hızlı ve uyumlu bir şekilde hazırlanabilmesi için

oluşturulmuştur. Microsoft tarafından geliştirildiği için, platform bağımlı çalışmaktadır. DirectX, grafik, ses, giriş/çıkış aygıtı yönetimi ve ağ üzerinden iletişim özellikleri konusunda oldukça yeteneklidir. Windows işletim sistemlerinde desteklenen bu kütüphane Linux'da ise aracı yazılımlar ile kullanılabilir [14].

2.2.3. Ogre3D

“Object-oriented Graphics Rendering Engine” kelimelerinin baş harflerinden oluşan Ogre3D açık kaynak kodlu olarak hazırlanmıştır. Massachusetts Institute of Technology (MIT) lisansına sahiptir ve platformdan bağımsız olarak çalışmaktadır. Nesneye dayalı bir mimari kullanılarak, Assaf RAMAN, Murat SARI ve dört arkadaşı ile beraber küçük bir takım tarafından geliştirilen ve büyük bir topluluk tarafından desteklenen Ogre3D, açık kaynak kodlu bir grafik gerçekleştirme motorudur. Güncel sürümü 1.9.2 RC2'dir. Oluşturulan modelleri önceden hazır bulunan fonksiyonları sayesinde bilgisayar ekranında gösterilmesini sağlayan bir C++ kütüphanesidir [15].

Ogre3D, gerçekleştirme işlemi için DirectX ve OpenGL kütüphanelerini kullanabilmektedir. Kullanmak istenilen kütüphaneye göre gerçekleştirme işlemi gerçek zamanlı (real-time) yapabilmektedir. Ogre3D vektör ve matris sınıfları, hafıza yönetimi vs. gibi farklı özelliklere de sahiptir. Kendi içerisinde ses, fizik ve ağ kütüphaneleri bulunmadığından, harici kütüphaneleri kullanmadan Ogre3D ile benzetim ve oyun geliştirmek mümkün değildir. Bu yüzden kullanılmak istenen kütüphaneleri dışarıdan eklemek gerekmektedir [16].

2.2.4. OpenSceneGraph

OpenSceneGraph (OSG) açık kaynak kodlu bir grafik programlama ara yüzüdür. En çok kullanıldığı alanlar; uçuş benzetimi, oyun, sanal gerçeklikler ve bilimsel sanallaştırmalar gibi yüksek performansta grafik uygulamaları olarak sıralanabilir. OSG kaynağının temeli OpenGL'dir. Bu sayede kütüphaneyi kullanan geliştiriciyi, düşük seviyeli grafik çağırımlarını uygulama ve uyumlaştırma gibi işlemlerden kurtarmaktadır. OSG Nesneye dayalı bir mimari kullanan framework ile

desteklemektedir. Geliştiricilerin daha hızlı geliştirme yapabilmeleri için kendi içerisinde birçok ek özelliği (parçacık, özel efektler, gölgelendirme, vb.) de barındırmaktadır [17].

2.3. BENZETİM

Benzetim, sadece teoride var olan veya gerçek hayatta kurulu olan bir sistemin özelliklerini taşıyan modelinin kurulması. oluşturulan benzetim modeli kullanılarak sistemin çalıştırılması, çalıştırma sonucunda elde edilen sonuçların gözlenmesi olmak üzere temel olarak üç aşamadan oluşur. Eğer benzetim işlemi, bilgisayar ortamında gerçekleştiriliyorsa bu tür benzetime bilgisayar benzetimi olarak adlandırılır. Benzetim gerçekleştirilirken gerçek sistemin matematiksel özellikleri ve bilgisayar grafikleri ile oluşturulan modelleri birleştirilerek bilgisayar ortamına aktarılır. Bu sayede sistem çalıştırılabilir hale getirilerek sistemin çıktıları izlenebilir. Benzetim kullanmanın yararları şöyle sıralanabilir [18];

- Sistemin benzetim yapılabilmesi için gerçek hayatta kurulu olma zorunluluğu yoktur.
- Benzetim sonuçlarını görmek için gerçek dünyada olan koşullara gerek yoktur.
- Parametre değişimlerine ani tepkiler verebilir.
- Gerçek sistem ile birlikte çalıştırılabilir.
- Model bir kere oluşturulduktan sonra sonsuz kere benzetim tekrarlanabilir.
- Ekonomiktir.

Günümüzde pek çok disiplinde ve alanda benzetim kullanılmaktadır [19-21]. Bilgisayar mimarilerinin gelişmesi ve işlemci kapasitelerinin artması sayesinde, çok daha karmaşık sistemler daha hassas bir biçimde modellenebilmektedir. Bu modeller ile oluşturulan diğer modellerle olan etkileşimleri daha yüksek bir doğruluk ile benzetilebilmesine olanak sağlamıştır. Benzetim sistemlerinin başlıca kullanım alanları; araştırma ve geliştirme, tasarım, eğitim, karar-destek ve eğlence alanları olsa da günümüzde tıp, otomasyon gibi farklı alanlarda da kullanılabilirler [22].

Benzetim uygulaması geliştirirken, fiziksel olayları hesaplamak için hazır kütüphaneler kullanılabilir. Bunlardan bazıları Bullet Physics, Newton Dynamics Engine, Open Dynamics Engine ve Physx olarak sıralanabilir. Gerçekleştirilecek benzetimin özelliğine bağlı olarak, bu fizik kütüphanelerinden biri veya bir kaç seçilir.

2.3.1. Bullet Physics

Bullet Physics, Erwin Coumans tarafından geliştirilmekte olan, açık kaynak kodlu bir fizik motorudur. İçinde çarpışma denetimi, katı cisim mekaniği, yumuşak cisim mekaniği ve dinamik fizik için gerekli olan fonksiyonlar mevcuttur. ZLib lisansı ile beraber ücretsiz olarak dağıtılmaktadır. C++ programlama dili ile beraber, Playstation 3, XBox 360, Wii, Windows, Linux, Mac OS, Android ve iPhone ortamlarında çalıştırılabilmektedir. Ayrık ve sürekli çarpışma denetimi sayesinde gerçekçi benzetilere imkân sağlamaktadır. Maya, Blender ve Collada ile beraber kullanılabilir [23].

2.3.2. Newton Dynamics Engine (NDE)

Newton Game Dynamics, Julio Jerez ve Alain Suero tarafından geliştirilen ücretsiz ve kapalı kaynak kodlu fizik motorudur. Bu kütüphane, oyunlarda ve diğer gerçek zamanlı uygulamalarda sahnede bulunan katı nesnelerin davranışlarını gerçekçi şekilde simüle eder. Bu kütüphane Windows, Mac OS, iPhone, ve Linux işletim sistemlerinde çalışabilir. Geliştirilen yeni boyutuyla beraber, çok çekirdekli işlemci ve grafik işlemcileri ile uyumlu hale gelmiştir. Akademik ve ticari birçok uygulamada kullanılmaktadır. C++ uygulama geliştirme ara yüzü ile beraber kullanılan bu kütüphane, kararlı ve güvenilir hızı ile temel fizik prensiplerini bilen kullanıcılar tarafından kolayca kullanılabilir [24].

2.3.3. Open Dynamics Engine (ODE)

Russel Smith tarafından 2000 yılının başında geliştirilmeye başlanan ODE, açık kaynak kodludur. ODE yüksek performans ile çalışan bir kütüphanedir. Katı cisim ve

esnek cisimleri simüle etmek için kullanılabilir. Platformdan bağımsız olarak çalışmaktadır. Kararlı sürüme geçmiş olan ODE, C/C++ ile kolayca kullanılabilir. Gelişmiş kemik sistemi ve sürtünme ile çarpışma tespiti entegrasyonu sayesinde gerçekçi benzetimler oluşturulmaktadır. Özellikle araç benzetimi uygulamalarında, sanal gerçeklik ortamlarındaki nesnelere ve sanal nesnelere benzetiminde kullanılmaktadır [25].

2.3.4. Physx

Ageia firması tarafından 2004 yılında geliştirilmeye başlanan gerçek zamanlı fizik motoru olan Physx, 2008 yılında Nvidia tarafından satın alınmıştır. Grafik işlemciler de üreten Nvidia Physx'in gelişimine büyük katkı sağlamaktadır. Fizik işlemlerinin daha hızlı gerçekleştirilmesi için PPU (Physics Process Unit) adı verilen genişleme birimi grafik kartlara eklenmiştir. Bu sayede fizik işlemlerini bu kart gerçekleştirmekte ve dolayısıyla donanımlar daha hızlı çalışmaktadır. Physx daha çok oyun geliştiricileri tarafından daha çok tercih edilmektedir. Microsoft Windows, Mac OS, Linux, PlayStation 3, Xbox 360 ve Wii ortamlarında çalışabilmektedir. Geliştiriciler için, hem ticari hem de ticari olmayan sürümleri mevcuttur [26].

BÖLÜM 3

BALİSTİK VE KATI CİSİM MEKANİĞİ

Bu çalışmanın teorik dayanakları balistik ve katı cisim mekaniği konularıdır. Balistik literatürde mermi bilimi olarak da adlandırılmaktadır. Temel olarak iç, dış ve terminal balistik olmak üzere üç bölümde incelenmektedir. Katı cisim mekaniği, katı olarak nitelendirilen malzemelerin davranışlarının incelendiği çalışma alanıdır.

3.1. BALİSTİK

Bu çalışmada, merminin hedef malzeme yüzeyi üzerine verdiği hasar incelenmiştir. Merminin namlunun içerisinde ortaya çıkardığı ilk enerjiden, hedefe ulaşmaya kadar geçen süre zarfındaki davranışını inceleyen bilim dalı balistikdir. Balistik konusu üzerinde çok uzun süredir çalışılmaktadır. Çin’de üretilen ilk baruttan günümüze kadar, balistik konusunda birçok akademik ve ticari ürün ortaya konulmuştur. Balistiğin temel çalışma konusu; merminin hedef üzerinde oluşturduğu nüfuz etme olgusudur. Çalışmaların bazıları bu zararı en az indirmek için; bazıları da bu hasarı en üst noktaya çıkarmak için yapılmaktadır [27].

Balistik inceledikleri olaylara göre iç balistik, dış balistik ve terminal balistik olmak üzere üç bölüme ayrılır.

3.1.1. İç Balistik

İç balistik genel olarak merminin ateşli silah içinde davranışlarını inceler. İç balistik, ateş edilmek üzere ateşli silaha yerleştirilen merminin ve silahın mekanik yapısını, çalışma prensibini, içerdeki patlamanın oluşumunu, mermi ve patlamanın mermi çekirdeği üzerindeki etkilerini ve silahı terk edene kadar gerçekleşen olayları inceler.

İç balistik yaklaşık 2 ila 4 milisaniyede gerçekleşen olaylar zinciridir [28]. İç balistik işlemini oluşturan basamaklar; merminin ateşlenmesi ile birlikte sevk barutu yanar. Bu yanma sonucunda, barutun içyapısında bulunan kimyasal enerji ortaya ısı enerjisi çıkarır. Isı enerjisi, mermi çekirdeğinin arkasındaki boşlukta bir basınç meydana getirir. Bu basınç mermiye uygulanır ve merminin ilk hızını almasını sağlar. Bu hız ile mermi namludan dışarıya çıkar [29].

3.1.2. Dış Balistik

Dış Balistik; merminin namlu ucundan çıktıktan sonra havadaki hareketini ve hedefe ulaşmaya kadar olan uçuşu esnasındaki davranışını inceler [28]. Temel konuları arasında; mermi dinamiği ve mermi çekirdeğinin uçuş dengesi, uçuşun yörüngesi ve havada geçen süre, hedefle olan temas noktası, temas açısı ve temas hızı yer almaktadır. Dış balistik, iç balistiğin devamı olarak nitelendirilebilir. Bu yüzden dış balistik konusunda çalışılırken, namlu ağzındaki mermi ilk hızının, dönme sayısının ve atış eğiminin bilinmesine ihtiyaç vardır. Bunların yanı sıra farklı mermi tipleri ve mermi şekilleri de bilinmelidir [30]. Merminin havada uçuşu sırasında; ortamın yoğunluğu, sıcaklığı, basıncı, viskozitesi ve rüzgâr gibi koşullar da etkilidir. Dolayısıyla bu koşulların da göz önünde bulundurulması gereklidir [29].

Dış balistikte mermi çekirdeğinin ilk çıkış hızı ve açısı bilinse dahi havada izleyeceği yörüngenin hesaplanması oldukça güçtür. Mermi çekirdeğine etki eden; sürüklenme, kaldırma, yunuslama gibi birçok kuvvet bulunmaktadır. Bu nedenle Aerodinamik hesaplamaları yapılarak mermi yörüngesi hesaplanmalıdır.

3.1.3. Terminal Balistik

Terminal balistik, merminin uçuşunun bittiği anda hedef ile temasa geçtiğinde mermi çekirdeğinin ve hedefin davranışlarını inceler. Mermiyi ateşlemenin temel amacı; hedefi vurmak ve hedef üzerinde bir hasar oluşturmaktır. Terminal balistiğin temel amacı da bu hasarı ve bu hasarı etkileyen faktörleri incelemektir. Terminal balistiğin temel konuları ise; mermi gövdesinin yüksek enerji taşıması sonucu hedef üzerinde parçalanması, kinetik ve kimyasal enerji sonucu hedefi delmesi ya da nüfuz etmesi,

sahip olduđu kimyasallar ile oluřturduđu patlama, oluřturduđu duman, parlama, dađılma v.b. dir [29].

Terminal balistiđinde, en sık kullanılan kelimelerden birisi olan, nřfuz etme kavramı; mermi ile hedef temas ettiđi anda merminin hedef üzerine ne řekilde nřfuz ettiđi olgusudur. Temas esnasında, mermi hedefe farklı řekillerde nřfuz edebilir. Mermi, hedef ile temasını tamamladıktan sonra dahi aynı yřnde bir hıza sahipse, yani geriye kalan son bir hızı varsa, “delinme” olayı geręekleřecektir. Merminin hedef ile temasını tamamlamaya hızı yetmeyip, hızı sıfırlanmıř ise buna “saplanma” adı verilir. Mermi hedef ile temasını tamamladıktan sonra ters yřnde bir hıza sahipse bu duruma “sekme” adı verilir [30].

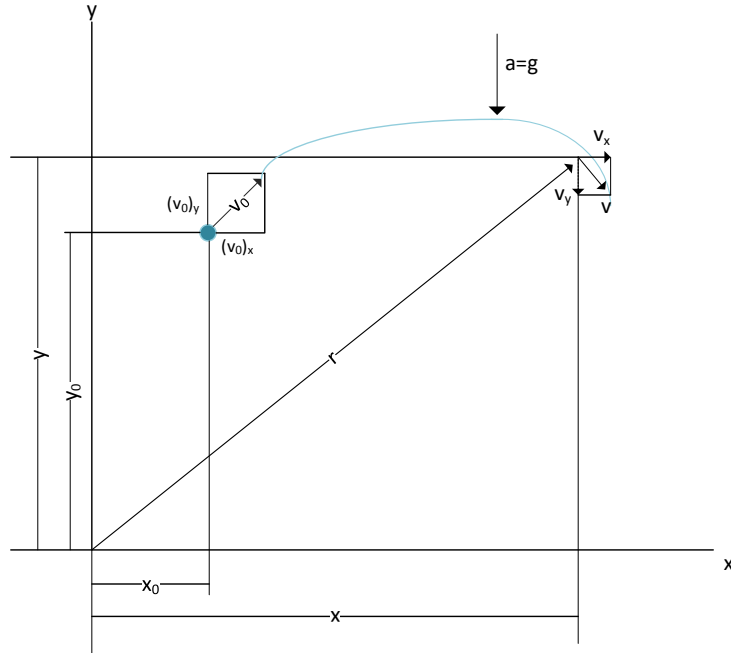
3.2. KATI CİSİM MEKANIĐİ

Mekanik ilgilendiđi alanlara gřre ęeřitli sınıflara ayrılabilir. Bu sınıflandırma, uđrařtıđı cismin fiziksel halini gřz řnřne alarak yapılabilir. Bu durumda mekanik temel olarak üç ana břlřme ayrılabilir. Bunlar; katı cisimlerin mekaniđi, sıvıların mekaniđi ve gazların mekaniđidir. Her bir břlřm kendi ięerisinde birden fazla alt břlřme ayrılabilir.

Katı cisimler eđer dıř yřk etkisi altında kalırlarsa, etkiye ve cisme bađlı olarak řekillerini deđiřtirirler. Bu deđiřmeyi etkileyen biręok faktřr vardır ve bu faktřrlerin her biri birer teknik problemdir. Bu teknik problemlerin ęözřmř yapıldıđında bu řekil deđiřtirmelerin oldukęa křçük olduđu gřrřlecektir. Bu yřzden řekil deđiřtirmeler biręok problemde ihmal edilirler. Bu ihmaller sonucu ortaya ęıkan cisme rijit cisim adı verilir. Rijit cisim, farklı dıř yřklere maruz kaldıđında, sahip olduđu iki nokta arasındaki uzaklıđı sabit kalan cisim olarak adlandırılabilir. Yani katı cismin ideal halidir. Bu ideal üzerinde ęalıřan mekaniđe rijit cisimlerin mekaniđi adı verilir [31].

3.2.1. Mermi Hareketi

Bir merminin serbest-uçuş hareketi, merminin ivmesi daima düşey doğrultuda olduğundan, çoğu kez dik bileşenler cinsinden incelenir. Şekil 3.1'de bir merminin eğik atışı temsil edilmektedir. Başlangıç değeri olarak (x_0, y_0) noktasından fırlatılan mermi incelendiğinde merminin alacağı yol, merminin başlangıç hızı v_0 'ın x ve y eksenindeki bileşenleri $(v_x)_0$ ve $(v_y)_0$ ile hesaplanır. Eğer hava direnci ihmal edilirse mermiye etkiyen tek kuvvet merminin ağırlığı olacaktır. Bu sayede yaklaşık $a_y = g = 9.81\text{m/s}^2$ 'lik aşağıya doğru sabit bir ivme kazanacaktır. O halde, $a_y = -g$ ve $a_x = 0$ olacaktır [32].

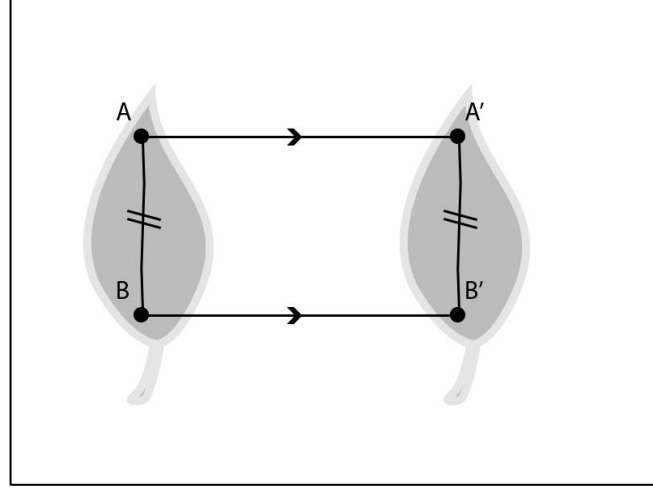


Şekil 3.1. Mermi hareket eğrisi [32].

3.2.2. Ötelenme ve Dönme

Bir katı cisim hareket ettirildiği süre aralığında katı cismin maddesel noktalarının hızları birbirine eşit ise bu harekete öteleme adı verilir. Hız $v = v(t)$, şeklindedir. Öteleme boyunca cismin maddesel noktalarının yörüngeleri birbirlerine paraleldir. Yani, cismin üzerinde bulunan her doğru hareket süresince orijinal doğruya paralel kalırsa bu hareket ötelemedir. Şekil 3.2'de görüldüğü gibi eğer bu paralellik doğru

boyunca devam ederse doğrusal öteleme, eğriler boyunca devam ederse eğrisel öteleme adı verilir [32].



Şekil 3.2. Ötelenme hareketi [32].

Bir cismin hareket ettirildiğinde sabit bir eksen etrafında dairesel yörünge izlerse bu harekete dönme adı verilir. Dönme hareketi boyunca cisimi oluşturan maddesel noktalar sabit bir eksen etrafında dairesel bir yörüngede hareket ederler. Bu hareket bir eksen boyunca gerçekleşiyorsa ve katı cismin o eksen üzerindeki tüm noktaların hızları sıfır ise, katı cisim bu eksen etrafında dönme hareketi yapıyor demektir. Dönmenin gerçekleştiği eksene de dönme eksenini denir [32].

3.2.3. Kuvvet ve İvme

Katı bir cismin hızı zamanla değiştiğinde, cismin ivmelenmekte olduğu söylenebilir. Bu kabulden yola çıkarak ivmenin, hızdaki değişimin ölçüsü olduğu anlaşılır. İvme ile çalışırken hızın büyüklüğü ile değil, zamanda olan değişimi ile ilgilenilir [33]. Newton'un ivme yasası olarak adlandırılan ikinci yasası, bir cismin üzerine uygulanan kuvvet, cismin sahip olduğu kütle ve cismin sahip olduğu ivme arasındaki ilişkiyi vermektedir. Bir cismin sahip olduğu ivme ve ona etki eden bileşke kuvvet (net kuvvet, $\sum F$) ile doğru orantılı; kütlesi ile ters orantılıdır.

$$\sum F = ma \quad (3.1)$$

F kuvvet, m kütle ve a ivmedir. SI birim sistemine göre kuvvetin birimi Newton'dur ve N harfi ile temsil edilir. Toplam kuvvet aşağıdaki denklemdeki gibi hesaplanır [33].

$$\sum F = F_1 + F_2 + F_3 + \dots + F_n \quad (3.2)$$

3.2.4. İş ve Enerji

Mekanikte bir iş yapılmış olabilmesi için, cismin uygulanan kuvvet doğrultusunda yer değiştirmiş olması gereklidir. Uygulanan F kuvveti, aynı yönde bir yer değiştirme sağlayabilirse, kuvvete maruz kalan cisim üzerinde iş yapmış olur. Fizikte iş yapmanın amacı enerji aktarımıdır. İş, bu alış-verişe verilen isimdir. Bir cisim için iş ve enerji ilkesini ifade eden denklem aşağıdaki gibidir [32].

$$\sum U_{1-2} = \frac{1}{2}mv_2^2 - \frac{1}{2}mv_1^2 \quad (3.3)$$

Bir sistemde bulunan enerji, kinetik ve potansiyel enerji gibi birden fazla türden olabilir. Toplam enerji sabit kalmak koşuluyla, sistemde bulunan enerji türleri arasında dönüşüm yapılabilir. Bir enerji türü artarken; diğer enerji türü azalacaktır. Diğer enerji türleri gibi sürtünme de ihmal edilirse, kinetik ve potansiyel enerjilerin toplamı sabit olacaktır [32].

$$\sum E_{toplam} = E_k + E_p \quad (3.4)$$

Eğer sistemde sürtünme varsa, toplam (mekanik) enerji (Kinetik ve Potansiyel Enerji toplamı) sabit olmayacaktır. Mekanik enerji zamanla azalacak, sürtünmeden dolayı ortaya çıkan ısı enerjisi ise artacaktır [32].

$$\sum E_{toplam} = E_k + E_p + E_{ısı} \quad (3.5)$$

3.2.5. İtme ve Momentum

İtme, bir cisme uygulanan kuvvet ve kuvvetin uygulanma süresinin çarpımıdır. Uygulanan kuvvet sabit veya zamana bağlı olabilir. Böyle durumlarda Newton'un kuvvet denkleminin integral formundan yararlanılır. Newton denklemi genel formu aşağıdaki gibidir [32].

$$F = m \frac{\Delta v}{\Delta t} \quad (3.6)$$

Momentum;

$$p = mv \quad (3.7)$$

Momentum denklemleri kullanılarak Newton denklemi aşağıdaki gibi yazılabilir.

$$F = \frac{\Delta p}{\Delta t} \quad (3.8)$$

Bu ifadenin integrali alınırsa aşağıdaki denklem elde edilir.

$$p_2 = p_1 + \int_{t_1}^{t_2} F dt \quad (3.9)$$

Burada p_1 , t_1 anındaki momentumu, p_2 de t_2 anındaki momentumu ifade etmektedir [34].

3.2.6. Malzeme Modeli

Benzetim ya da diğer uygulamalarda malzemelerin davranışının kullanılabilmesi için modellerinin oluşturulması gerekmektedir. Malzemenin davranışı modellenirken Oxley, Johnson-Cook, Zerilli-Armstrong gibi modellerden istifade edilebilir [35]. Oxley, deneysel sonuçlara dayanarak, malzemenin deformasyonunu ampirik olarak modellemiştir [36]. Zerilli-Armstrong yüksek hızlı çarpışma ve bazı metal (Al, v.b.) şekillendirme için bir model geliştirmiştir [37]. Johnson ve Cook ise yüksek

gerilmeye sahip malzemeler için bir model geliştirmiştir. Bu modeli diğerlerinden ayıran en önemli özelliği yüksek sıcaklık ve yüksek şekil değiştirme hızlarında malzemenin davranışını yüksek doğrulukla tanımlamasıdır. Modelin denklemi, denklem 3.10'da verilmiştir [38].

$$\bar{\sigma} = [A + B\varepsilon^n][1 + C \ln \dot{\varepsilon}^*][1 - T^{*m}] \quad (3.10)$$

Burada A, B, C, n ve m malzemenin sabit değerleri olup, deneysel yöntemler ile bulunmuştur. Bu denklemde ε , plastik gerinme eşdeğerini temsil etmektedir. $\dot{\varepsilon}$, plastik gerinim hızını temsil etmektedir ve

$$\dot{\varepsilon}^* = \frac{\dot{\varepsilon}}{\dot{\varepsilon}_0} \quad (3.11)$$

formülü ile bulunur. $\dot{\varepsilon}_0$ değeri 1.0 s^{-1} alınmaktadır. T^{*m} değeri, boyutsuz sıcaklık değeri olup,

$$T^{*m} = \frac{T - T_r}{T_m - T_r} \quad (3.12)$$

denklemini ile bulunur. Bu denklemde T_r oda sıcaklığını, T_m malzemenin ergime sıcaklığını ifade etmektedir [38]. Bu tez çalışmasında kullanılan malzemelerin akma mukavemeti değeri ANSYS LS-Dyna yazılımında Johnson-Cook modelini kullanılarak elde edilen sonuçtan temin edilmiştir.

BÖLÜM 4

ÖNCEKİ ÇALIŞMALAR

Yapılan çalışmanın iki ana unsuru vardır. Bunlar malzeme bilimi ve bilgisayar grafikleridir. Bu konular ile ilgili literatürde pek çok çalışma mevcuttur. Bu çalışmalardan önemli olduğu düşünülenler bu bölümde özetlenmiştir.

Şenel ve arkadaşları “İleri kompozit zırh malzemelerin balistik analizi” adlı çalışmalarında ANSYS LS-Dyna programını kullanarak bir merminin seramik ve metal tabakalı zırhın üzerinde oluşturduğu etkinin sayısal analizini ve sonrasında balistik darbe testini gerçekleştirmişlerdir [39].

Borvik ve arkadaşları “Normal and oblique impact of small arms bullets on AA6082-T4 aluminium protective plates” adlı çalışmalarında bir merminin yüzeylere farklı açılarla çarpması durumunda meydana getirdiği etkiyi sonlu elemanlar metoduyla incelemişlerdir. Borvik ve arkadaşları bu çalışmalarında, malzeme yüzeyi olarak 20 mm kalınlığındaki AA6082-T4 alüminyum malzeme kullanmışlardır. Çalışmada kullanılan merminin başlangıç hızı 830 m/s olup, çelik çekirdeğe sahiptir. Merminin, malzeme yüzeyine 0°, 15°, 30°, 45° ve 60° açılarıyla çarptığında meydana gelen etkiyi Johnson-Cook modeli kullanarak açıklamışlardır [40].

Borvik ve arkadaşları “Perforation resistance of five different high strength steel plates subjected to small-arms projectiles” adlı çalışmalarında ise iki farklı mermi tipi ve beş farklı malzeme yüzeyi kullanmışlardır. Mermi tipi olarak BR6 ve BR7 sınıfı (7.62x51 mm), malzeme yüzeyleri olarak Weldom 500E, Weldom 700E, Hardox 400, Domex Protext 500 ve Armox 560T alaşımları kullanılmıştır. Benzetim Ansys LS-Dyna programı kullanılarak hazırlanmıştır. Benzetim hazırlanırken Johnson-Cook ve Cockcroft-Latham metodlarını kullanmışlardır [41].

Juraj Hub “Numerical simulations of the 12.7-PZ32 bullet penetration of the aluminium alloy sheet metal” adlı çalışmasında, uçak gövdelerinde kullanılan alüminyum malzeme üzerine 12.7 mm merminin vurması sonucu oluşacak hasarı sayısal olarak simüle etmiştir. Kullanılan malzeme yüzeyinin kalınlığı 1.2 mm’den 2.5 mm’ye kadar çıkmaktadır. Vuruş açısı 90°’ye ayarlanmış olup, merminin başlangıç hızı 811.1 - 821.3 m/s arasındadır. Bu benzetim 2B düzlemde gerçekleştirilmiştir. Hub aynı çalışmayı deneysel olarak da icra edip elde ettiği sonuçları nümerik sonuçlarla karşılaştırmıştır [42].

Hub ve arkadaşları “Ballistic limit evaluation for impact of pistol projectile 9 mm luger on aircraft skin metal plate” adlı çalışmalarında ise uçak gövdesinde kullanılan yüksek mukavemetli alüminyum 2024-T3 ve 9 mm kalibrelik mermi ile çalışmışlardır. Ansys Autodyn programı kullanarak 2B benzetim gerçekleştirmişlerdir. Benzetim için SEM kullanmışlardır. SEM, çoğu mühendislik alanında kullanılan problemi alt bölümlere ayırma ve her bir bölümde fonksiyonun ifadesini polinom olarak seçerek çözüm arayan yöntemdir. Benzetim sonucunda da malzeme yüzeyinin balistik limiti elde edilmiştir [43].

Özşahin ve Tolun “Alüminyum levhaların yüksek hızlı çarpma davranışları için ampirik bir model” adlı çalışmalarında; AA2024-T31 alaşımı levhaların yüksek hızlı çarpma yüklemesi durumundaki delinme miktarını üstel olarak ifade etmişlerdir. Bu üstel ifade ile elde edilen sonuçları, önceki çalışmaları ile karşılaştırıp ifadenin doğruluğunu belirtmişlerdir [44].

Pechoucek, Rolc, Buchar “Fragment simulating projectile penetration into layered targets” isimli çalışmalarında, yüksek mukavemetli çelik tabakaları kullanarak deney gerçekleştirmişlerdir. 20 mm çapında ve 1000 m/s hızında bir mermi ile deneyi gerçekleştirmişlerdir. Deneysel verilerden yararlanarak sayısal bir model oluşturup, bu modeli Ansys LS-Dyna yazılımı ile benzetim gerçekleştirilmiştir [45].

Merminin verdiği zarar ile ilgili bilgisayar grafiği alanında yapılan çalışmalara örnek olarak Mazarak ve arkadaşları “Animating exploding objects” adlı çalışması verilebilir. Mazarak ve arkadaşları bu çalışmalarında, şok dalgasının grafiksel olarak

nasıl tasvir edileceği açıklamış ve matematiksel modeli sunmuşlardır. Patlama sonucunda, hacimsel tabanlı fiziksel nesnelere gerçekçi olarak patlama efektini yansıtmaktadırlar. Bu işlem uygulanırken önemli sorunları belirtmişler ve işlemi benzetim olarak göstermişlerdir [46].

Terzopoulos ve Witkin "Physically Based Models with Rigid and deformable components" adlı çalışmalarında deforme olabilen malzemeler ve katı cisimler için bir model geliştirmişlerdir. Bu modelde deformasyon işlemi referans noktaya göre ayrıştırma yoluyla yapılmaktadır. Bu sayede deformasyon işlemini geometrik olarak temsil edecek eşitlik elde etmişlerdir [47].

Nealen ve arkadaşları "Physically Based Deformable Models in Computer Graphics" adlı çalışmalarında deforme olabilen malzemelerin fiziksel temelli olarak modelleme işleminde gerçekleşen önemli çalışmaları derlemişlerdir. Gerçekçi animasyonlar ve benzetimlerinin hazırlanması, SEM, voksel modelleme gibi alanlardaki gelişmeleri sunmuşlardır [48].

Kaufman "Volume Visualization: Principles and Advances" adlı çalışmasında hacimsel imge oluşturma teknikleri, hacimsel imgeleri kaplama teknikleri, hacimsel yüzey modelleme konuları hakkında bilgi vermiştir. Ayrıca özel amaçlı hacimsel imgelerin kullanımı hakkında bilgi sunmuştur [49].

Terzopoulos ve Fleischer "Modeling inelastic deformation: viscoelasticity, plasticity, fracture" adlı çalışmalarında katı olmayan nesnelere fiziksel olarak benzetimi ve animasyonunu gerçekleştirmişlerdir. Esnek olmayan materyaller üzerinde deformasyon uygulaması gerçekleştirilmiştir. Bu uygulama görsel ve matematiksel olarak sunulmuştur [50].

Metaxas ve Terzopoulos "Dynamic Deformation of Solid Primitives with Constraints" adlı çalışmalarında parametrik olarak tanımlanan katı modellerden dinamik modeller oluşturmuşlar ve bu modellere lokal SEM ile deformasyon uygulamışlardır. Ayrıca modelde bulunan kısıtlar için bir metot geliştirmişlerdir. Bu metot sayesinde dinamik modeller ile beraber, kısıtlar kullanılarak animasyon

oluřturmuřlardır [51].

BÖLÜM 5

MERMİ HASAR SİMÜLASYONU

Bu çalışmada farklı tipteki mermilerin, farklı malzeme yüzeyleri üzerine verdikleri hasar modellenmiş ve bilgisayar grafikleri ile benzetimi gerçekleştirilmiştir. Bu uygulama geliştirilirken ilk olarak modeller ve bu modellerin sahip olduğu animasyonlar Blender programı ile hazırlanmıştır. Hazırlanan modeller Ogre3D ile görüntülenmek üzere dışarı aktarılmıştır. Ogre3D ile modeller ve animasyonları ekranda gösterilmiştir. Modellere katı cisim özelliği kazandırılarak benzetim işlemi gerçekleştirilmiştir. Benzetim işlemi gerçekleştirilirken Bullet Physics isimli fizik kütüphanesi kullanılmıştır.

5.1. MODELLERİN HAZIRLANMASI

Uygulamada kullanılacak modellerin gerçekteki karşılıkları ile aynı olması için modeller tasarlanırken parametrik özellikleri birbirlerinin aynısı olacak şekilde hazırlanmaya çalışılmıştır. Modeller açık kaynak kodlu 2B/3B tasarım hazırlanabilen Blender programı kullanılarak tasarlanmıştır. Uygulamada mermi, hedef yüzey, hedef tahtası ve karakter modelleri bulunmaktadır. Bu modeller için hedef tahtası ve karakter öğelerinin tasarımı herhangi bir parametreye bağlı kalmadan tasarlanması gerekirken, mermi ve hedef yüzey ile ilgili gerçek parametrik değerlerin kullanılması gereklidir.

Uygulamada birbirinden farklı 5 tipteki hafif silah mühimmatı grubunda yer alan mermi türü kullanılmıştır. Benzetim aşamasında kullanılacak olan parametreler Çizelge 5.1’de yer almaktadır. Bu mermi verileri Türk Ordusuna Silah ve mühimmat üretimi yapan MKE resmi web sitesinden alınmıştır.

Çizelge 5.1. Kullanılan mermi tipleri ve parametreleri [52].

Mermi Grubu (mm)	Kütle (g)	Başlangıç hızı (m/s)	Uzunluk (mm)	Genişlik (mm)
7.62 x 51	~24.5	838.8	71.12	0.6
5.56 x 45	~12.5	914.4	57.4	0.7
9 x 17	~8.4	270	25	0.3
9.65	~15.7	180	30.6	1
12.7 x99	~114	872	138.4	1.27

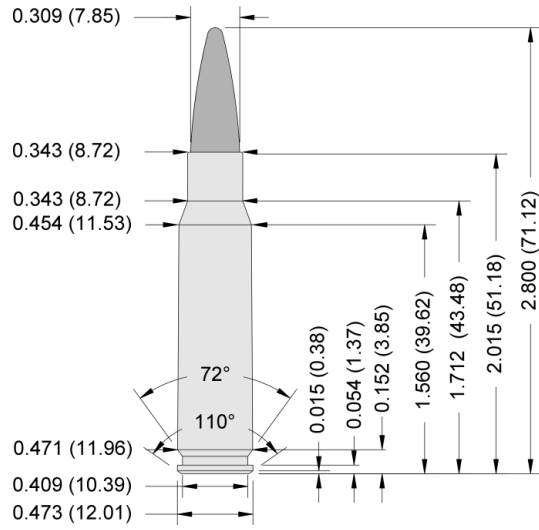
Hedef olarak kullanılacak malzeme yüzeyleri için 5 farklı materyal türü seçilmiştir. Bu malzemelerin parametrik değerleri Çizelge 5.2’de yer almaktadır.

Çizelge 5.2. Kullanılan malzeme yüzeyleri ve parametreleri¹.

Malzeme Yüzeyi	Akma Gerilmesi (MPa)	Kütle (1cm ³) (g)
Alüminyum (Al)	290	2.73
Demir (Fe)	175	7.86
Kurşun (Pb)	80	11.37
Çelik 4340 (AISI4340)	792	7.85
Gümüş (Ag)	50	10.5

Bu çalışmada kullanılan modeller Blender isimli 3B grafik programı ile modellenmiştir. Modeller oluşturulurken gerçek ölçüleri dikkate alınmıştır. Şekil 5.1’de 7.62 x 51 bir merminin gerçek ölçüleri görülmektedir.

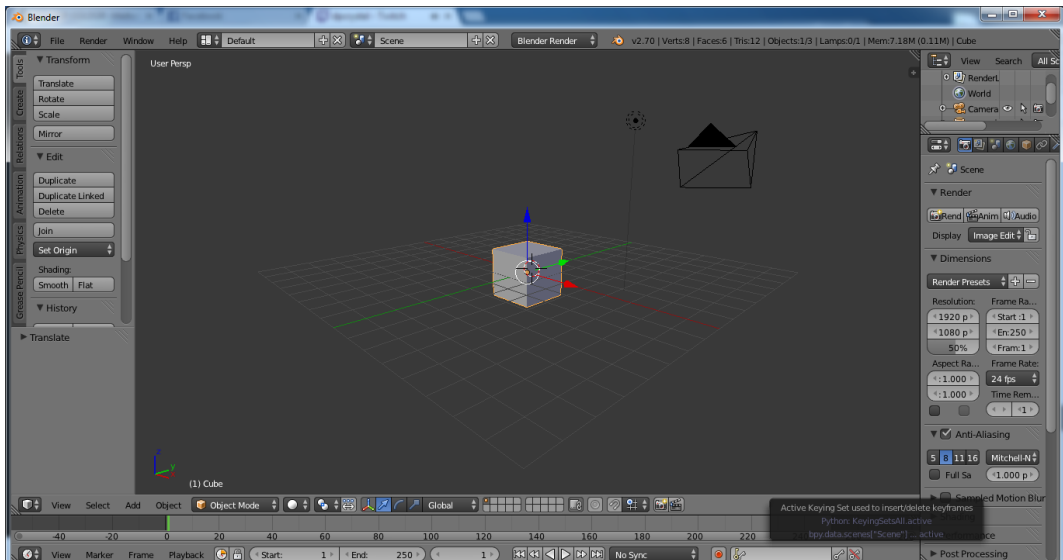
¹ Bu veri Ansys programında kullanılan referans verileridir.



Şekil 5.1 7.62 x 51 mm Nato mermi ölçüleri [53].

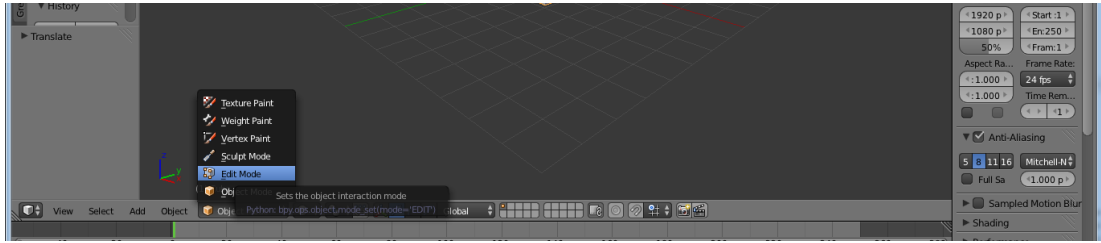
5.1.1. Mermi Modelleme

Modellenecek nesnenin ölçüleri belirlendikten sonra modelleme için Blender programı açılır. Program açıldığında Şekil 5.2'deki ekran gelecektir. Sahnede bir adet küp bulunmaktadır. Modelleme yapılırken, genellikle modeller bu küp üzerinde çıkarma (extrude), ölçekleme (scale), döndürme (rotate), öteleme (translate) işlemleri uygulanarak oluşturulur. Örnek olarak bu çalışmada kullanılan modellerinin başlangıcı küp nesnesidir.



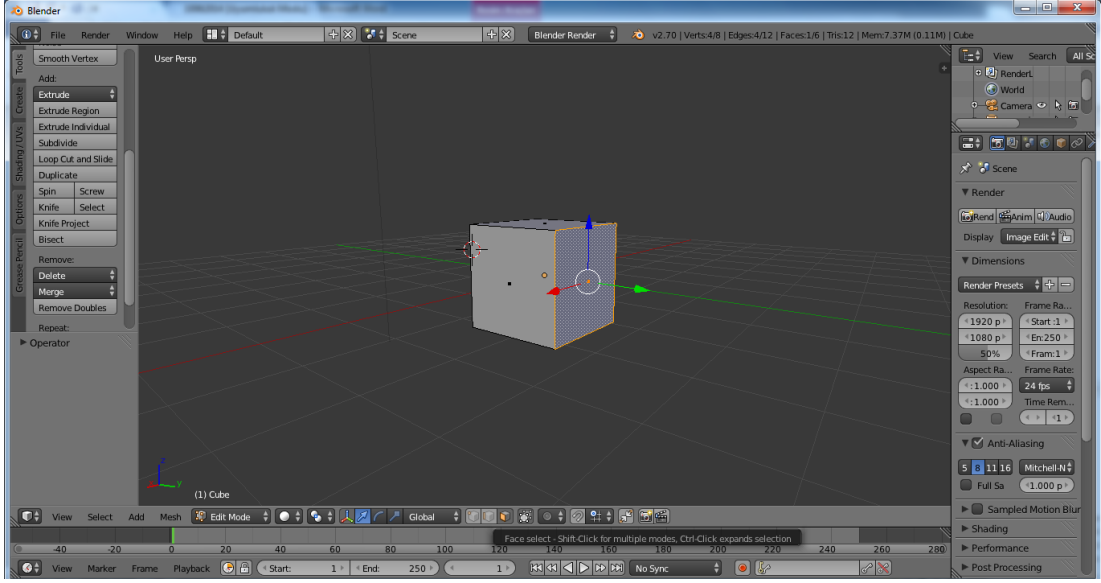
Şekil 5.2. Blender açılış ekranı.

Blender içerisinde modelleme yapabilmek için birden fazla mod bulunmaktadır. Şekil 5.3'de görüldüğü gibi bunlar Kaplama Boyama (Texture Paint), Ağırlık Boyama (Weight Paint), Nokta Boyama (Vertex Paint), Heykeltraş Kipi (Sculpt Mode), Düzenleme Kipi (Edit Mode) ve Nesne Kipi (Object Mode) modlarıdır. Blender ile çizim yapılırken gerektiğinde bu kipler arasında geçiş yapılarak istenilen işlem gerçekleştirilir. Örneğin model üzerinde düzenleme (çıkarma, ölçekleme, vs.) yapabilmek için Düzenleme Kipine geçiş yapılır. Nesne Kipi ile birden fazla katman arasında geçiş yapmak için kullanılır. Örneğin bir sahnede birden fazla karakter bulunuyor ise bu karakterler arası geçiş bu kip ile sağlanır.



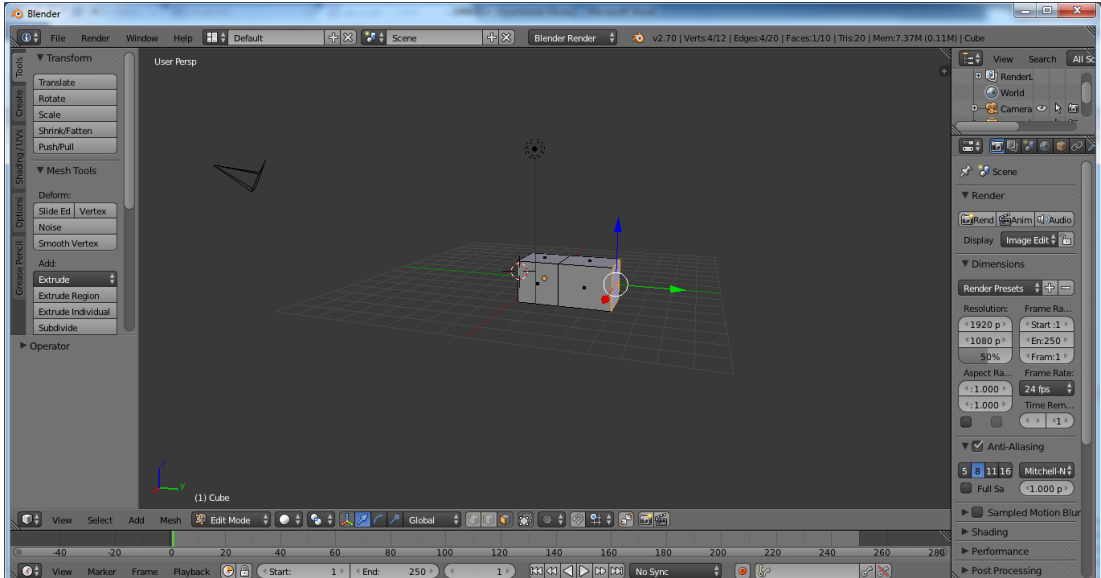
Şekil 5.3. Blender modları seçim ekranı.

Düzenleme kipine geçildiğinde modelin köşe noktaları, kenar çizgileri ya da yüzeyleri üzerinde işlem yapılabilir. Modelin hangi özelliği üzerinde işlem yapılacaksa Blender programında o özellik seçili olmalıdır. Şekil 5.4'deki modelde yüzeyler üzerinde işlem yapılması istenildiği için modelin yüzeyleri seçilebilir hale gelmektedir. Bir seçim yapıldığında, o yüzeyi oluşturan kenarlar ve noktalar seçilecektir.



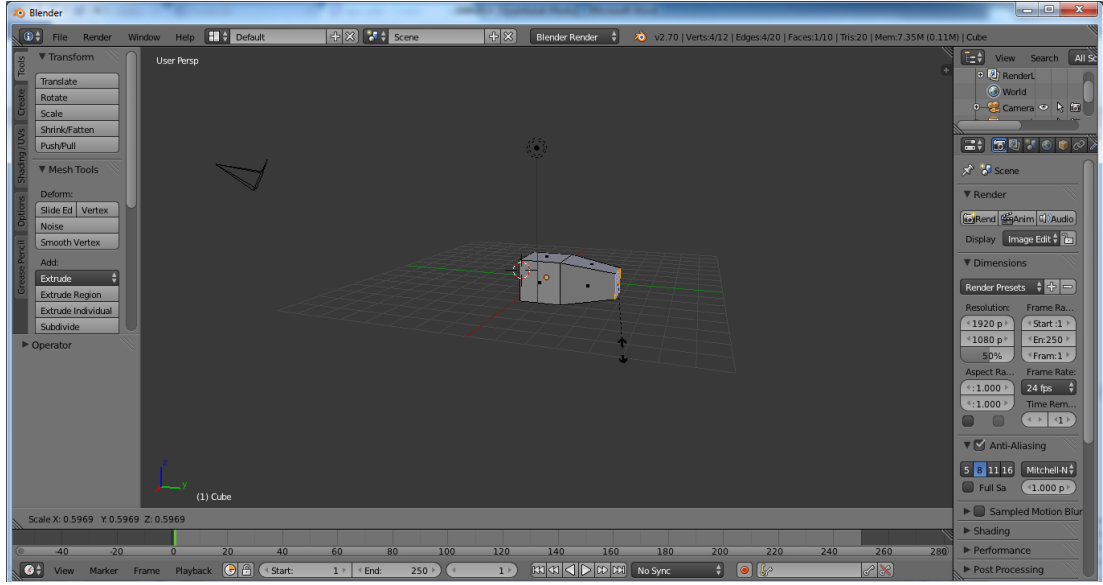
Şekil 5.4. Düzenleme kipi yüzey seçimi.

Şekil 5.5’de gösterilen modele çıkarma işlemi uygulanmıştır. Çıkarma işlemi modelin seçili olan özelliğini (nokta, kenar, yüzey) içeri ya da dışarı taşıma şeklinde gerçekleşir. Fakat bu taşıma işlemi sonunda yeni yüzey ya da kenarlar eklenir. Şekil 5.5’de dışarı çıkartım işleminin sonucu görülmektedir. Bir yüzey seçilip çıkartım işlemi uygulandıktan sonra çıkartım işlemi uygulanan yüzey ile diğer yüzeyleri birleştirmek için otomatik olarak yüzeyler eklenecektir.



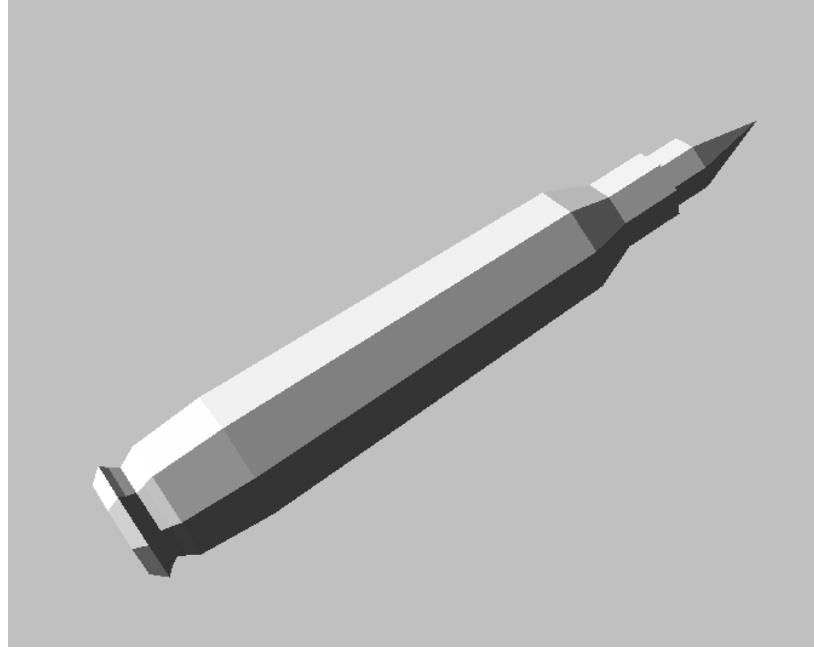
Şekil 5.5. Çıkartım işlemi uygulanmış küp modeli.

Bir modele birden fazla işlem uygulanabilmektedir. Önce öteleme, sonra çıkartım sonra da ölçeklendirme uygulanabilir. Tasarlanacak modele göre bu işlemler gerçekleştirilir. Şekil 5.6'da çıkartım uygulanmış modele ölçeklendirme işlemi de uygulanmıştır. Bu sayede merminin uç kısmının modeli oluşturulur. Ölçeklendirme yapılırken bilgisayarın faresi ileri geri hareket ettirilerek ya da klavye ile tam değer girilerek ölçeklendirme işlemi gerçekleştirilir.



Şekil 5.6. Ölçeklendirme işlemi uygulanmış model.

Bu işlemler sayesinde bir mermi oluşturulabilmektedir. Çıkartım, ölçekleme gibi birçok işlem sayesinde oluşturulan mermi modeli Şekil 5.7'de görülmektedir.

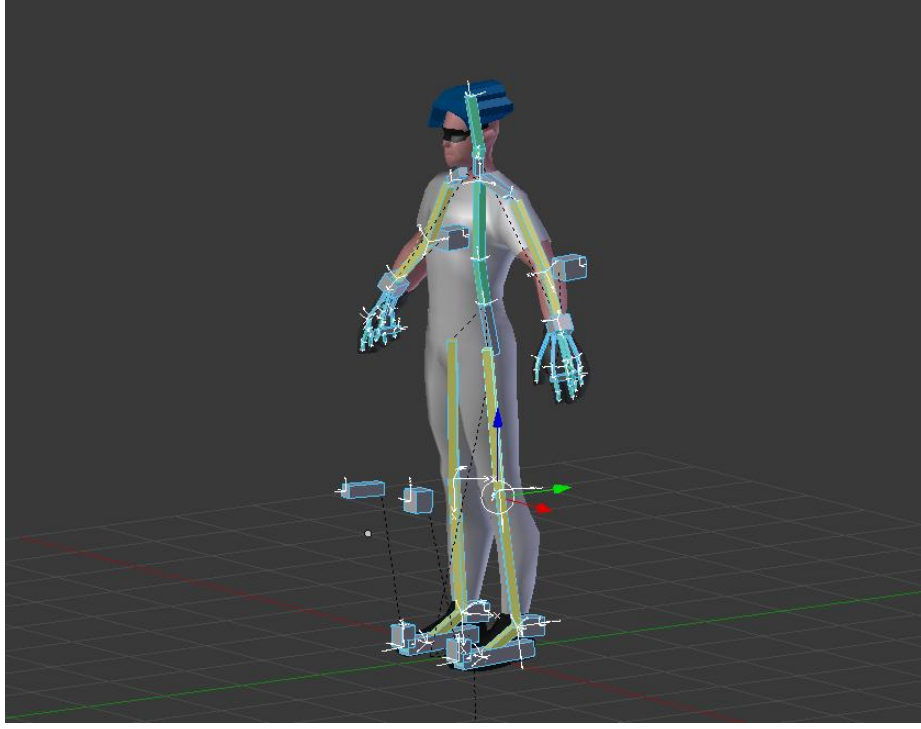


Şekil 5.7. 7.62 x 51 NATO mermi 3 boyutlu görünümü.

5.1.2. Karakter Modelleme

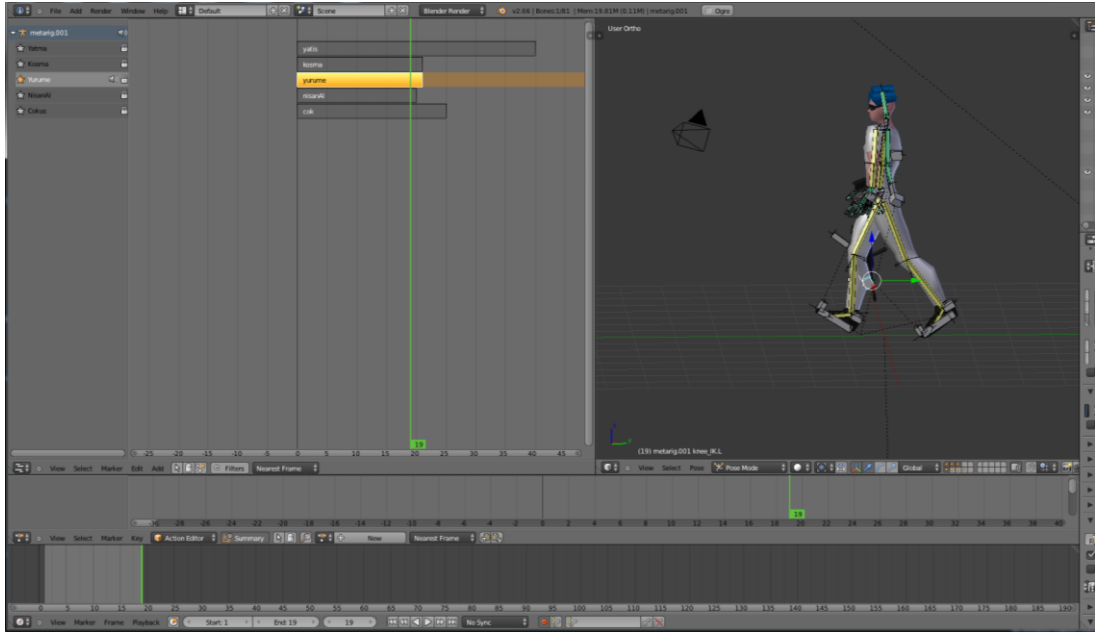
Uygulamada kullanılmak üzere modellenen karakter ise Şekil 5.8’de görülmektedir. Bir karakter modeli, yüzeyler, yüzey kaplamaları, kemik sistemi ve animasyonlardan oluşur. Kemik sistemi eklemek için ise Nesne Kipine geçilmelidir. Ekle menüsü altında Armature menüsünün içerisinde normal kemik bulunmaktadır. Buradan seçim yapılarak modele kemik eklenebilir. Bu çalışmada modelleme işlemi yapılırken karakter modeline kemik ve eklem tanımlamaları yapılmıştır. Modeldeki kemik ve eklemler sayesinde daha gerçekçi animasyonlar yapılabilmektedir.

Bir model üzerinde animasyon gerçekleştirmek, modelin gerekli noktalarında konum değiştirmek olarak düşünülebilir. Tüm bu noktasal değişim model dosyası içerisinde saklanmaktadır. Aynı anda çok fazla noktasal değişim olduğundan, sadece noktaları değiştirerek animasyon geliştirmek zor olabilir. Bu yüzden kemik sistemini kullanarak animasyon geliştirmek, hem görsel açıdan daha gerçekçi olacak hem de tasarımcıya büyük kolaylıklar sağlayacaktır. Kemik sisteminde bulunan ters kinematikler sayesinde, insan vücudundaki kemik sistemi bilgisayar ortamında modellenebilir.



Şekil 5.8. Karakterin 3 boyutlu ve kemik sistemli modellenmiş hali.

Uygulamada kullanılan modelin; nişan alma, yere yatma, diz çökme, koşma ve yürüme animasyonları hazırlanmıştır. Animasyon ekranı şekil 5.9’da görülmektedir. Her bir animasyon ayrı ayrı çerçeveler yardımıyla hazırlanmıştır.



Şekil 5.9. Animasyon ekranı.

5.2. MODELLERİN GÖRSEL GERÇEKLEME İÇİN HAZIRLANMASI

Blender ile hazırlanan 3B modellerin, ekran sahnesinde temsil edilebilmesi için modelin bir ağa (mesh) dönüştürülmesi gerekir. Genellikle grafik hazırlamak için geliştirilen kütüphanelerde modellerin bu tip dönüşümlerini gerçekleştiren eklentiler bulunur. Bu dönüşümler ile bir ekran sahnesi üzerinde farklı tür ve özelliklere sahip modeller kolayca görüntülenebilir. Bununla birlikte Blender ile üretilen modellerin kullanılabilmesi için birkaç işlemin daha yapılması gerekmektedir.

Ogre3D kütüphanesinin kullandığı dosya formatı mesh'tir. Bu yüzden kullanılacak modelin mesh biçimine dönüştürülmesi gereklidir. Bu dönüşüm iki adımda gerçekleşmektedir. İlk adımda; Blender ile çizilen modeller Blender2Ogre isimli bir eklenti ile xml formatında dışarıya aktarılabilir. Bu işlem sonrasında modellerin materyal bilgilerini (parlaklık, renk, kaplama) bulunduran “.material” uzantılı materyal dosyası ve modelin konum ve yüz bilgilerinin bulunduğu xml dosyası oluşturulur. Materyal dosyası Ogre3D içerisinde bulunan uygun klasöre (media klasörü içerisindeki materials klasörü) aktarılır.

İkinci adım, XML formatındaki dosyanın, Ogre3D kütüphanesinin mesh formatına çevrilmesidir. XML dosyasının mesh haline çevrilmesi için Ogre3D ile beraber gelen Ogre3DXMLConverter programından yararlanır. Bu sayede, modelin sahip olduğu tüm özellikler (animasyon, kemik sistemi vs.) mesh formatına dönüşür. Böylece Şekil 5.10'da görüldüğü gibi modeller görsel gerçeklemeye hazır hale gelir.

```
H:\Users\Rafet\Desktop\OgreCommandLineTools_1.7.2>OgreXMLConverter merni.mesh.xml
1
--- OPTIONS ---
source file      = merni.mesh.xml
destination file = merni.mesh
log file        = OgreXMLConverter.log
interactive mode = false
lod levels      = none (or use existing)
Generate edge lists = 1
Generate tangents = 0
semantic       = TRANGEMI
parity         = 0
split mirror   = 0
split rotated  = 0
Reorganise vertex buffers = 1
Optimise animations = 1
--- END OPTIONS ---

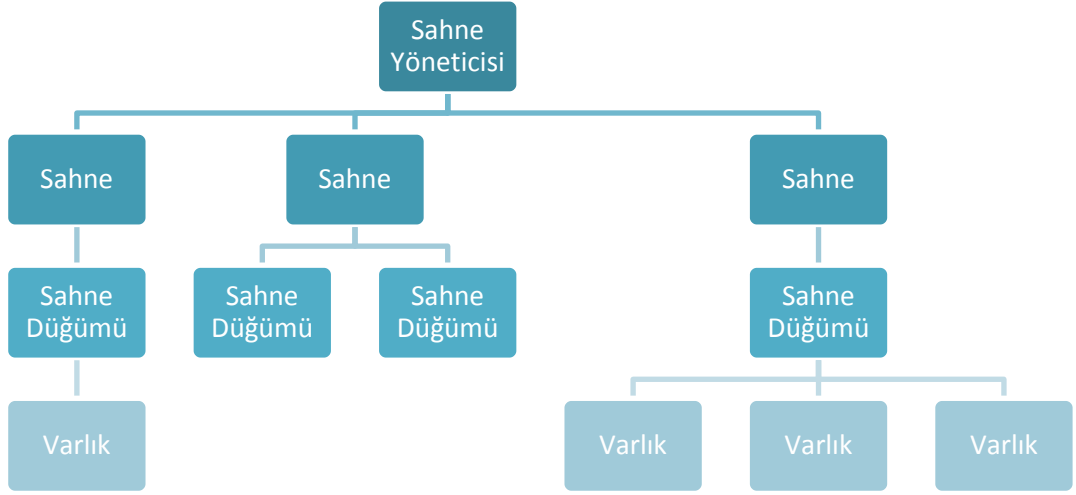
Creating resource group General
Creating resource group Internal
Creating resource group Autodetect
Registering ResourceManager for type Mesh
Registering ResourceManager for type Material
Registering ResourceManager for type Skeleton
XMLMeshSerializer reading mesh data from merni.mesh.xml...
Reading geometry...
Geometry done...
Reading submeshes...
Submeshes done.
Reading mesh names...
Mesh names done.
XMLMeshSerializer import successful.
Reorganising vertex buffers to automatic layout..
Generating edge lists...
MeshSerializer writing mesh data to merni.mesh...
File header written.
Writing mesh data...
Writing submesh...
Exporting submesh texture aliases...
Submesh texture aliases exported.
Submesh exported.
Exporting bounds information...
Bounds information exported.
Exporting submesh name table...
Submesh name table exported.
Exporting edge lists...
Edge lists exported.
Mesh data exported.
MeshSerializer export successful.
Unregistering ResourceManager for type Skeleton
Unregistering ResourceManager for type Material
Unregistering ResourceManager for type Mesh

H:\Users\Rafet\Desktop\OgreCommandLineTools_1.7.2>
```

Şekil 5.10. OgreXMLConverter işlem ekranı.

5.3. GÖRSEL GERÇEKLEME

Ogre3D, nesneye dayalı olarak geliştirilmiş bir grafik kütüphanesidir. Bu yüzden görsel gerçekleştirme sırasında her bir model nesne ile temsil edilir ve “varlık” (entity) olarak adlandırılır. Bir görüntü içindeki bütün öğeler sahne grafi kullanılarak ekrana aktarılır. Bir sahne grafi düğümler ve varlıklardan teşkil olmuştur. Graf üzerindeki düğümler ile dönüşüm işlemleri (yer değiştirme, döndürme, ölçekleme gibi) organize edilir. Ekranda yer alan her bir model, bir varlık nesnesi olarak tanımlanmıştır. Her bir varlık bir sahne düğümüne bağlı olmalıdır. Örneğin birden fazla sahne, sahne düğümü ve varlığın bulunduğu bir sahne grafi Şekil 5.11’de gösterilmiştir.



Şekil 5.11. Ogre3D sahne grafi.

Şekil 5.12’de örnek bir sahne görülmektedir. Sahnede hedef tahtası, karakter ve yer olmak üzere 3 adet tanımlanmış varlık bulunmaktadır. Her bir varlığın kendine özgü bir düğümü bulunmaktadır. Bu sayede birbirlerinden ayrı olarak hareket ettirilebilmektedirler. Fakat kamera karakter ile beraber hareket etmektedir. Bu yüzden kamera ve karakter aynı sahne düğümünde olursa, beraber hareket ettirilmeleri daha kolay olacaktır. Sahnede bulunan gökyüzü ise bir varlık değil, sahnenin bir özelliği olarak tanımlanmıştır.



Şekil 5.12. Varlıkların sahne üzerinde görünümü.

Ogre3D'de sahne, görüntülenme işleminin gerçekleştiği ortamdır. Tüm varlıklar bu ortam üzerinde yer alır. Sahneyi yönetmek için Ogre::SceneManager isimli bir sınıf bulunmaktadır. Bu sınıftan varsayılan olarak bir adet sahne yöneticisi türetilmiş olup, adı mSceneMgr olarak tanımlanmıştır. İstenilmesi halinde createSceneManager isimli fonksiyon çağırılarak yeni bir sahne yöneticisi oluşturulabilir.

```
Ogre::SceneManager* Sahne;  
Sahne = mRoot->createSceneManager(Ogre::ST_GENERIC, "sahneAdi");
```

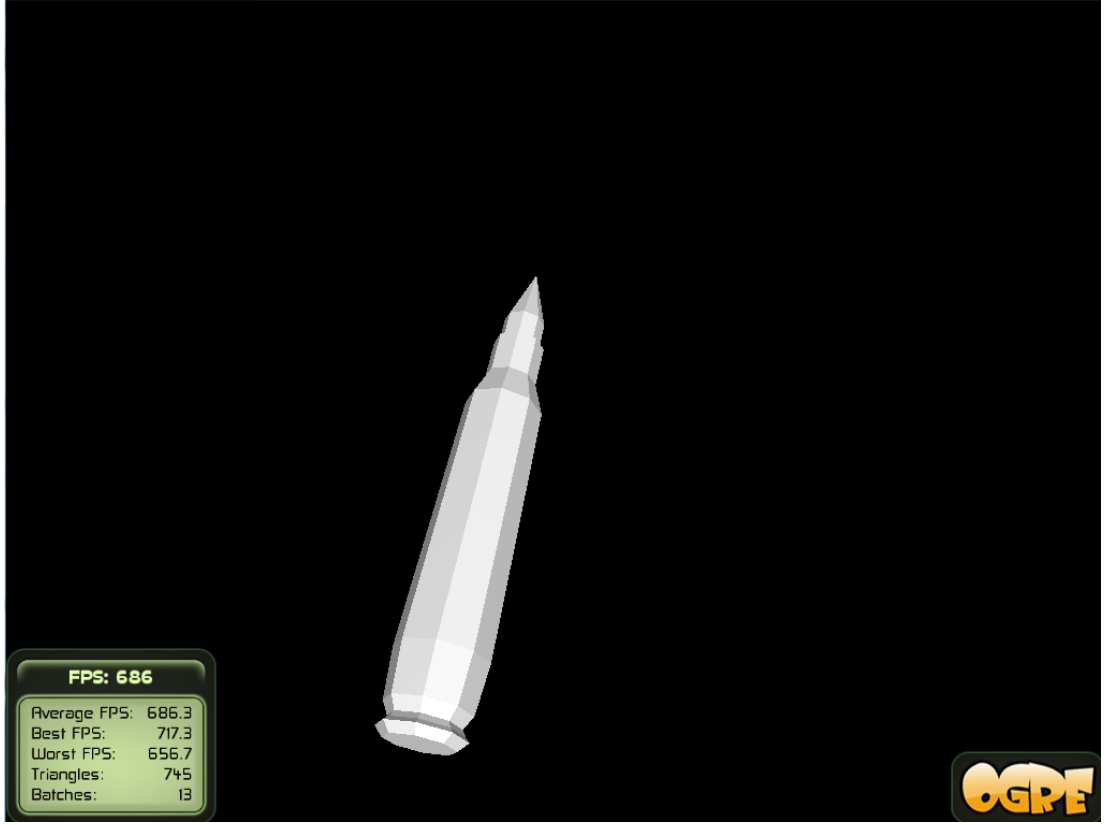
Sahne grafi üzerinde bulunan sahne düğümlerini tutan sınıf Ogre::SceneNode sınıfıdır. Bu sınıf sayesinde varlıklar düğüme eklenerek ekranda gösterilebilirler. Yeni bir sahne düğümü eklemek için kök sahne düğümünün altına yeni bir sahne düğümü eklenebilir. Bu sayede tüm sahne düğümleri kök düğüm altında yer alır.

```
// Yeni bir sahne düğümü oluştur  
Ogre::SceneNode* mermiNode = mSceneMgr->getRootSceneNode() -  
>createChildSceneNode("mermiNode");
```

Ogre3D ile uygulama geliştirirken, ekranda görüntülenecek her bir varlık bir sahne düğümü içerisinde yer almalıdır. Sahne düğümü üzerinde yer almayan varlıklar ekranda görüntülenmezler. Aynı şekilde her bir sahne düğümü de bir sahneye bağlanmalıdır.

Ogre3D ile varlıklar Ogre::Entity isimli sınıfta tutulur. Yeni bir varlık oluşturmak için createEntity isimli fonksiyon çağrılır. Yeni bir sahne eklenmediği takdirde, varlıklar mSceneMgr isimli değişken kullanılarak sahneye eklenirler. Aşağıdaki kod satırları eklenip uygulama çalıştırdıktan sonra şekil 5.13'deki görüntü elde edilecektir.

```
// Yeni bir varlık oluştur.  
Ogre::Entity* mermi = mSceneMgr->createEntity("mermi",  
"mermi.mesh");  
  
// Yeni bir sahne düğümü oluştur  
Ogre::SceneNode* mermiNode = mSceneMgr->getRootSceneNode() -  
>createChildSceneNode("mermiNode");  
// Yeni varlığı sahneye ekle  
mermiNode->attachObject(mermi);
```



Şekil 5.13. Ogre3D ile merminin görsel gerçekleştirilmesi.

Sahneye eklenen varlığa döndürme, ölçeklendirme, öteleme gibi birçok işlem uygulanabilir. Bu işlemler sahne düğümünün gerekli fonksiyonları (rotate, translate, scale, vs.) ile gerçekleştirilir. Bu sayede sahne üzerindeki yerleşim ayarlanabilir.

```
// Sahne düğümünü 0,0,0 noktasına ötele.  
mermiNode->translate(Ogre::Vector3(0,0,0));  
// Sahne düğümünü x ekseninde 0.5 radyanlık açı ile döndür.  
mermiNode->rotate(Ogre::Vector3::UNIT_X,Ogre::Radian(0.5));  
// Sahne düğümünü x,y ve z eksenin 10 kat daha büyüt.  
mermiNode->scale(Ogre::Vector3(10,10,10));
```

Ogre3D ile esas görüntü kameralar sayesinde ekrana aktarılır. Kameraların konumu ve baktığı açı değiştirilmesi sayesinde kullanıcı kendisini sahne üzerinde hissedebilir. FPS (First Person Shooter) olarak adlandırılan oyun grubunda karakterin göz hizasına yerleştirilen bir kamera sayesinde kullanıcının kendisi oyunun içerisindeymiş hissi verilmeye çalışılmaktadır. Bu işlemin daha kolay yapılabilmesi için kamera ve karakterin aynı sahne düğümüne eklenerek birlikte hareket ettirilmesi gerekmektedir.

```

//Asker Karakterini ve Onun Kamerasını Ekle
AnaNode = mSceneMgr->getRootSceneNode()-
>createChildSceneNode("AnaNode",Ogre::Vector3(0,2,0));
AskerNode = AnaNode->createChildSceneNode("AskerNode");
KameraNode = AnaNode->createChildSceneNode("KameraNode");
KameraPitchNode = KameraNode->createChildSceneNode("KameraPitchNode");
KameraPitchNode->attachObject(mCamera);
askerEntity = mSceneMgr->createEntity("Asker", "Asker.mesh");
AskerNode->yaw(Ogre::Degree(180.0f));
AskerNode->attachObject(askerEntity);
AskerNode->scale(3,3,3);
AskerNode->translate(0,8,150);

```

Blender veya diğer 3B modelleme programları ile hazırlanan modelin, Ogre3D ile gerçekleştirilecek hale geldiğinde animasyon bilgileri de modelin mesh dosyası içerisinde bulunmaktadır. Ogre3D animasyonları otomatik olarak tanır ve kendi içerisinde bulunan animasyon durum yönetimi (AnimationState) değişkeni ile kontrol eder. Bunun için yapılması gereken sadece modelin animasyon bilgisini mAnimationState isimli değişkene eklemek ve özelliklerini (Sürekli mi çalışacak?, modele olan etkisi, başlat/durdur) belirlemek olacaktır.

```

//Askerin yatis isimli animasyonunu ekle
mAnimationState = askerEntity->getAnimationState("yatis");
//Sadece bir kez tekrarla
mAnimationState->setLoop(false);
//Animasyonu başlat
mAnimationState->setEnabled(true);
//Modele sadece bu animasyon etkisin
mAnimationState->setWeight(1);

```

5.4. SİMÜLASYON

Günlük hayatta gerçekleşen olayların benzetimi sadece görüntüleme işlemleri ile sonuçlandırılmaz. Görüntülemenin yanında grafiksel öğelerin fiziksel durumları ve reaksiyonlarının da benzetilmesi gerekir.

5.4.1. Fiziksel İşlemler

Ogre3D'nin fizik kütüphanesine sahip olmaması nedeni ile, geliştirilen uygulamada fiziksel durumların (yer çekimi, uygulanan kuvvet, vs.) hesaplanması ve simüle edilebilmesi için Bullet Physics isimli kütüphane kullanılmıştır. Bullet Physics kütüphanesi açık kaynak kodlu ve ücretsiz olması; C++ dilini kullanması ve Ogre3D ile uyumlu çalışması gibi özelliklerden dolayı tercih edilmiştir.

Sahne üzerinde fiziksel özelliklere sahip olacak olan varlıklara bu özellikler Bullet Physics isimli fizik kütüphanesi kullanılarak atanmıştır. Bu kütüphanenin Ogre3D ile beraber kullanımını kolaylaştırmak amacıyla OgreBullet isimli bir ara kütüphane bulunmaktadır. Bu ara kütüphane sayesinde fiziksel işlemlerin gerçekleştirileceği benzetim dünyası projeye eklenmiştir. Tüm fiziksel olaylar bu dünya üzerinde gerçekleşmektedir. Bu dünya her çerçevenin başlangıcında ve bitişinde güncellenmektedir. Bu işlem stepSimulation fonksiyonu ile gerçekleşmektedir.

```
//Fiziksel dünyayı güncelle
mWorld->stepSimulation(evt.timeSinceLastFrame);
```

Bu dünya üzerinde yer alan fiziksel nesnelere birden fazla tipte (Katı cisim, yumuşak cisim, sabit nesne vs.) tanımlanabilirler. Fiziksel nesnelere çarpışma tanımlama işlemi için de fiziksel şekli bulunmalıdır. Bu sayede diğer fiziksel nesnelere olan teması sonucunda fiziksel işlemler gerçekleşecektir. Katı cisim özelliklerine sahip bir fiziksel nesne oluşturmak için OgreBulletDynamics::RigidBody sınıfı kullanılmaktadır.

```
OgreBulletDynamics::RigidBody *cekirdekBody = new
OgreBulletDynamics::RigidBody(
    "mermiCekirdegi" +
    Ogre::StringConverter::toString(mNumEntitiesInstanced),
    mWorld);
```

Oluşturulan nesneye bir çarpışma şekli atanabilmesi için öncelik bu çarpışma şeklinin OgreBulletCollisions::CollisionShape ana sınıfından türetilmesi gerekmektedir. Çarpışma şekilleri birden fazla tipte (küp, silindir, koni vs.) olabilmektedir. Fakat bazı durumlarda birden fazla tipin tek bir çarpışma şekli olarak tanımlanması gerekmektedir. Böyle durumlarda birleşik çarpışma şekli (compound) oluşturulabilir. Örneğin, merminin çarpışma şeklinde bir adet silindir ve koni bulunmaktadır.

```
//Çarpışma Şeklini Belirle
Ogre::Vector3 axe = Ogre::Vector3(0,0,1);
//Silindirik çarpışma şekli
OgreBulletCollisions::CylinderCollisionShape* cekirdekSilindir = new
OgreBulletCollisions::CylinderCollisionShape(Ogre::Vector3(0.6f,0.5f,0.5f),axe);
```

```

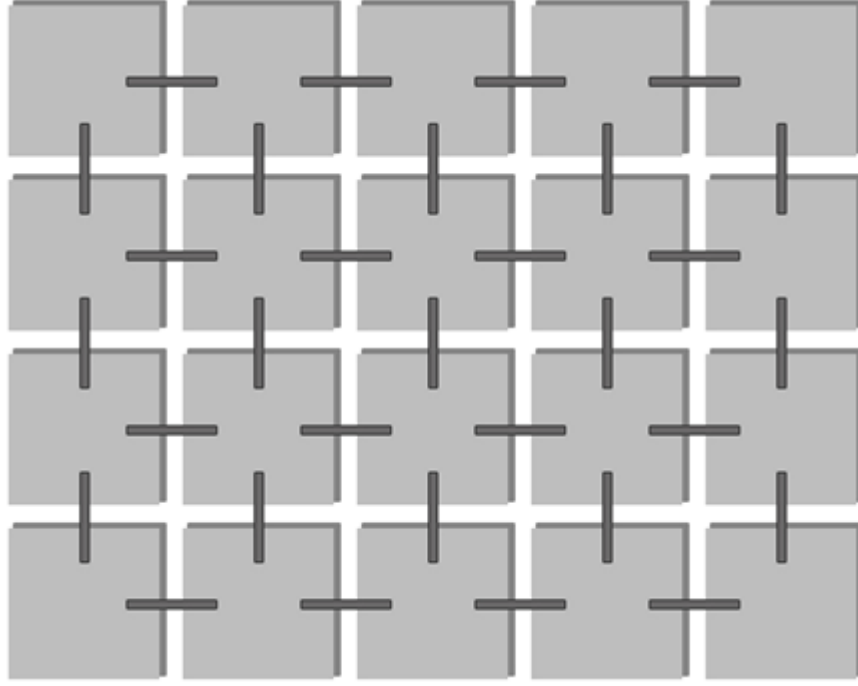
//Koni çarpışma şekli
OgreBulletCollisions::ConeCollisionShape* cekirdekKoni = new
OgreBulletCollisions::ConeCollisionShape(0.6f,2.0f,axe);
//İkisini birleştirmek için ana şekli tanımla
OgreBulletCollisions::CompoundCollisionShape* cekirdekShape = new
OgreBulletCollisions::CompoundCollisionShape();
//Silindirik çarpışma şeklini birleşime ekle
cekirdekShape->addChildShape(cekirdekSilindir);
//Koni çarpışma şeklini birleşime ekle
cekirdekShape->addChildShape(cekirdekKoni,Ogre::Vector3(0,0,-
2),Ogre::Quaternion(0,0,-1,0));

```

Çarpışma şekli setShape fonksiyonu ile fiziksel nesneye atanmaktadır. Fiziksel şekil atandıktan sonra nesne fiziksel olaylardan (örneğin varsa yer çekimi gibi) etkilenmeye başlar. Nesneye istenilirse başlangıç hızı da setLinearVelocity fonksiyonu ile 3B olarak verilebilir.

Bu çalışmada, delinme ve parçalanma vokseller yardımıyla sağlanmaktadır. Voxel(voxel), hacim kazanmış piksel olarak tanımlanabilir. Bu çalışmadaki vokseller çarpışma şekline sahip olan fiziksel nesnelere dir. Çarpışma şekli olarak küp çarpışma şekline sahiptirler. Şekil 5.14’de hedef yüzeyi için kullanılan vokseller görülmektedir.

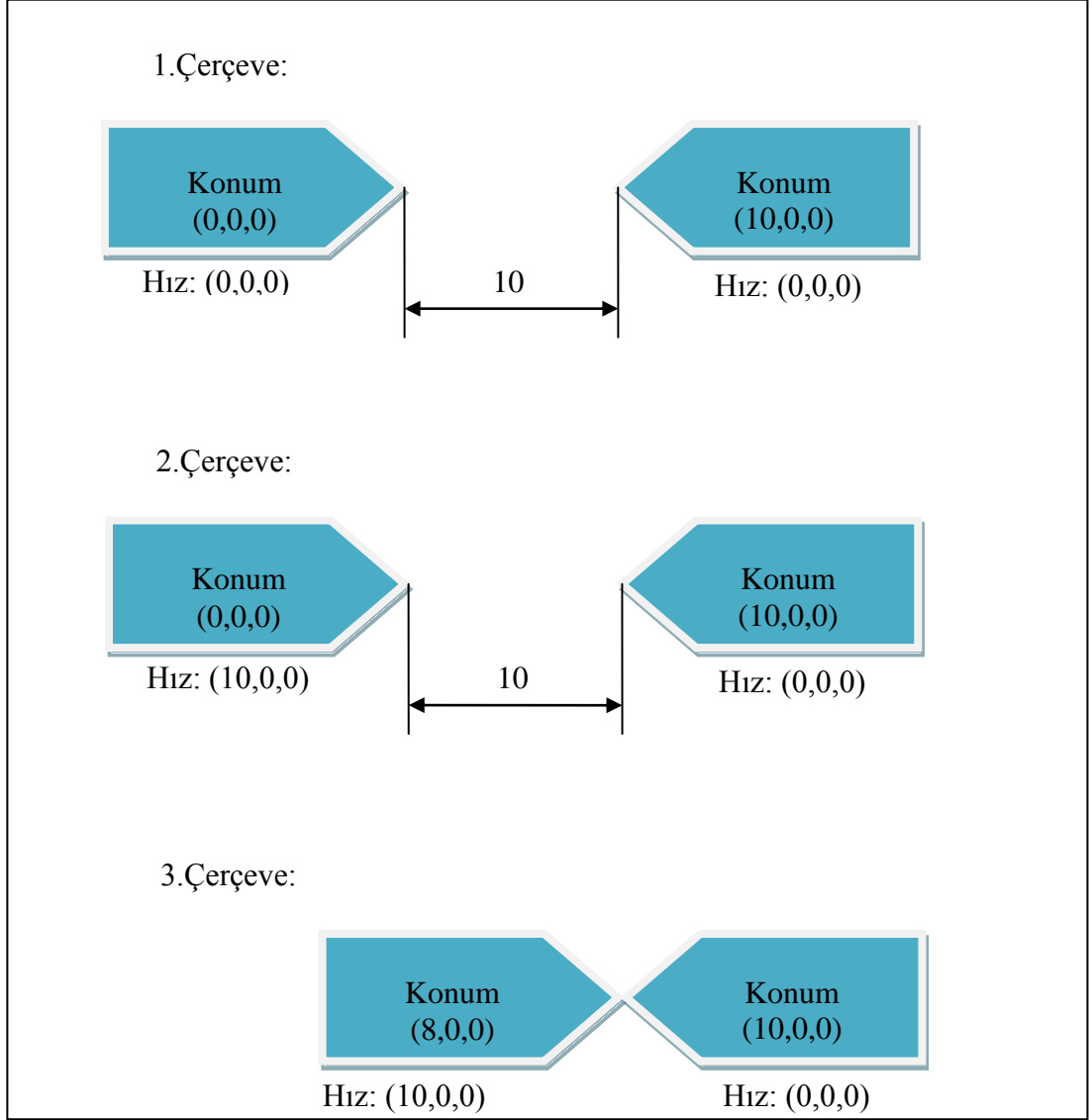
Vokseller kısıt (constraints) olarak adlandırılabilen özellikler içeren bağlar ile birbirlerine bağlıdır. Fiziksel olayların tatbiki sonucu vokseller arasındaki bağlar etkilenecek ve ilgili nesnenin deforme olup olmadığı tespit edilecektir. Her bir voxel 4 adet sabit nokta kısıtı ile diğer voksellere bağlanmıştır. Bir kısıt nesnesi eşik değeri, konum bilgisi ve bağlı olduğu görüntü nesnesi verileri içerir. Vokseller ve vokseller arası kısıtlara ait eşik değeri deforme için kullanılan parametredir. Her bir kısıta etki eden itme kuvveti malzemenin eşik değerinden büyük ise o kısıt kopacaktır. Voksele ait tüm kısıtların kopması halinde ise ilgili voxel serbest kalacaktır. Voxel, üzerine uygulanan kuvvet yönüne bağlı olarak harekete geçecektir. Vokselin hareket etmesi, malzemenin delinmesi anlamına gelmektedir.



Şekil 5.14. Voksel yapısı.

Hedef yüzeyin deformasyona uğramasının ilk şartı bir çarpışmaya maruz kalmasıdır. Benzetim sistemlerinde, çarpışma tespiti (collision detection) olarak isimlendirilen kontrol ile varlıkların birbirleriyle temas edip etmediği test edilir. Çarpışma tespitinin yapılabilmesi için her bir varlığın bir çarpışma şekline (collision shape) sahip olması gereklidir.

Bu çalışmada belirli bir hızda atılan merminin hedef yüzeye çarpma tespitinin yapılması gereklidir. Çarpışma durumunun tespitinin yanı sıra, çarpma hızının vokseller arasındaki bağlara etkisi diğer bir önemli noktadır. Zira bağları koparacak bir kuvvet ile çarpışma gerçekleşmiş ise bu durumda hedef yüzeyin görüntüsü değişmek zorundadır. Bu yüzden bilgisayar grafiklerinin benzetimde gerçek zamanlı olarak temsil edilebilmesi için çarpışma kontrolü, çarpışmanın gerçekleşmesinden önceki benzetim çerçevesinde tespit edilmesi gerekmektedir. Çarpışma testinin senaryosu Şekil 5.15'deki gibi kontrol edilmektedir.



Şekil 5.15. Çarpışma kontrolü şeması.

Birinci çerçevede verilen durumda iki nesne arasındaki mesafe 10 birimdir. Her iki nesnenin de hızları sıfır olduğundan dolayı sonraki çerçevede söz konusu nesnelerin konumları değişmeyecektir. Bu durumda aralarındaki mesafe de değişmeyecektir. Bu yüzden sonraki çerçevede aralarındaki mesafe sıfırın altına inmeyeceği için çarpışma yoktur denilir.

İkinci çerçevede ilk nesnenin başlangıç hızı x eksenini için 10 birim olarak verilmiştir. İki nesne arasındaki mesafe de 10 birimdir. Buna göre bir sonraki çerçevede iki nesne birbiri ile temas edeceği için çarpışma olacaktır.

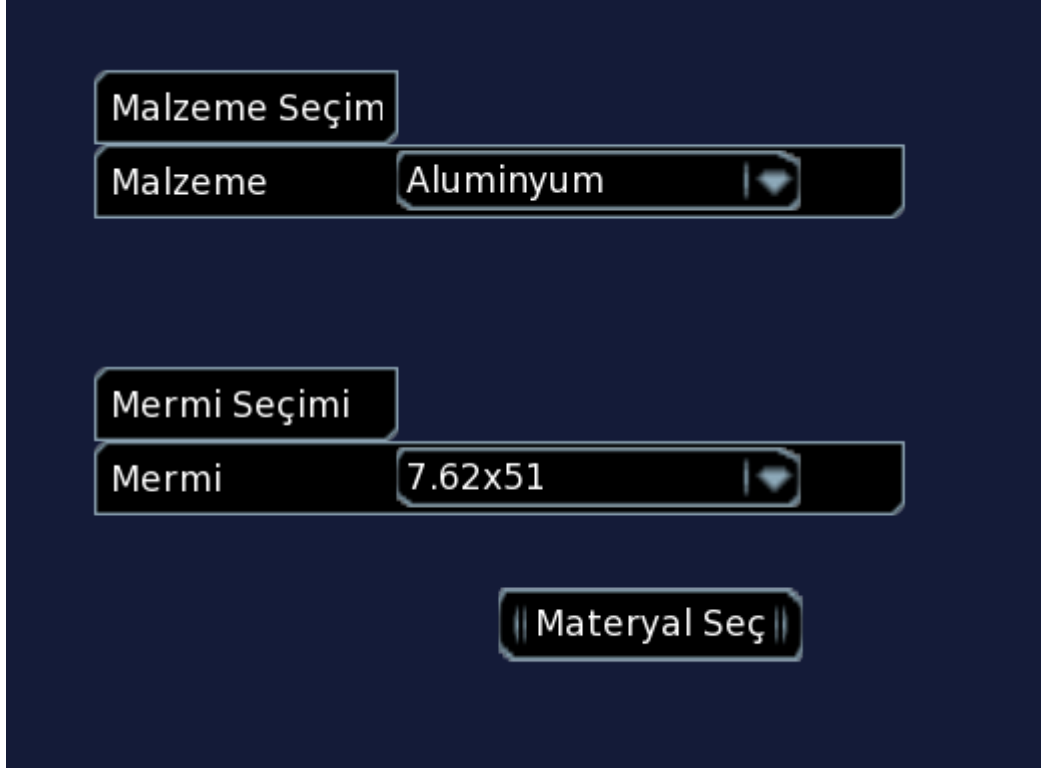
Üçüncü çerçevede hızlara bakılmaksızın aralarındaki mesafe sıfır birim olduğu için söz konusu nesnelere arasında çarpışma olacağı tespit edilebilir.

Bu algoritmada temel amaç hareket eden nesnenin sahip olduğu hızı da hesaba katarak, sonraki adımda hangi noktada olacağını, dolayısıyla çarpışmanın olup olmayacağını önceden tespit etmektir. Daha sonra delinme ve parçalanma işlemleri gerçekleşir.

Bullet Physics kütüphanesinin kendi fonksiyonları kullanılarak birden fazla çarpışma kontrolü sağlamak mümkündür. Fakat ekranda çok fazla fiziksel nesne olduğundan ve bu noktaların her birinin birbirleriyle olan ilişkilerinden dolayı, tüm nesnelere arası çarpışma kontrolü yapılmalıdır.

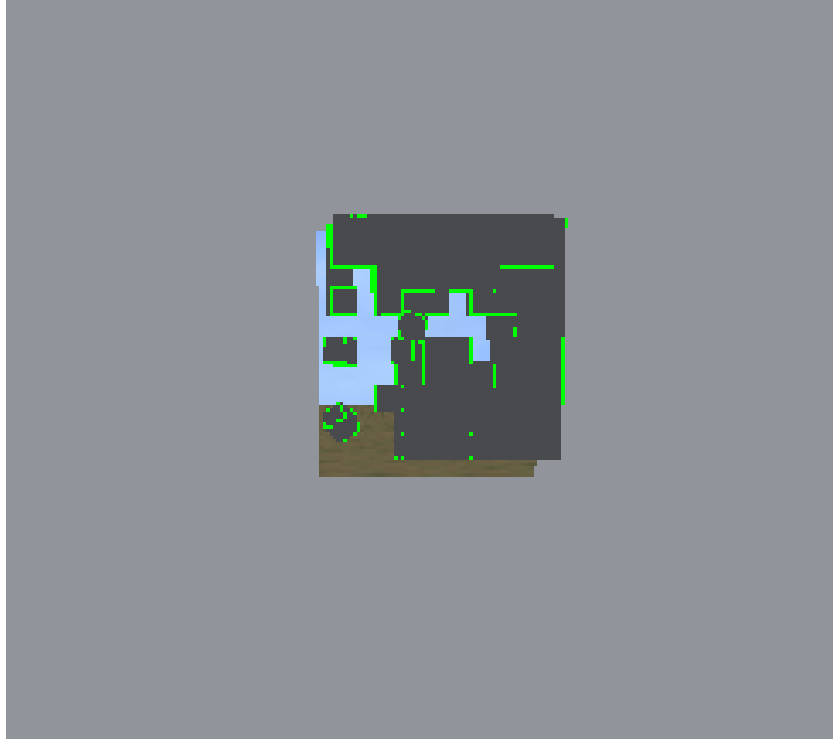
5.4.2. Grafikselle İşlemler

Çarpışma işlemi gerçekleştiğinde, vokseller çarpışma noktasını merkeze alarak sahneye eklenmektedir. Bu voksellerin kaplamaları dinamik olarak malzemeye bağlı olarak eklenmektedir. Malzeme ve mermi seçim ekranı Şekil 5.16'daki gibidir. Bu seçim ekranı CEGUI isimli kütüphane kullanılarak yapılmıştır. Seçim gerçekleştiğinde, dinamik olarak kaplamalar ve diğer parametreler değişecektir. Vokseller ve kısıtların eşik değerleri seçili olan malzemenin eşik değerleri olacaktır.



Şekil 5.16. Malzeme ve mermi seçim menüsü.

Hedef yüzeyi mesh yüzeyle temsil edilmektedir. Bu yüzey dinamik olarak oluşturulmuştur. Bu sayede, hafızadan yer kazanımı sağlanmıştır. Ayrıca yüzeyler dinamik olarak oluşturulup, dinamik olarak belirli alanları çıkarılabilmektedir. Mermi hedefe çarptığı anda, hedef olarak tanımlanmış olan örgü (mesh) yüzeyinde voksellerin oluşturulacağı alan çıkarılıp, vokseller o bölgeye yerleştirilmektedir. Şekil 5.17’de gösterildiği gibi o bölgedeki vokseller hareket ettiğinde delinme işlemi, grafiksel olarak gösterilmektedir.



Şekil 5.17. Hasar almış malzeme yüzeyi.

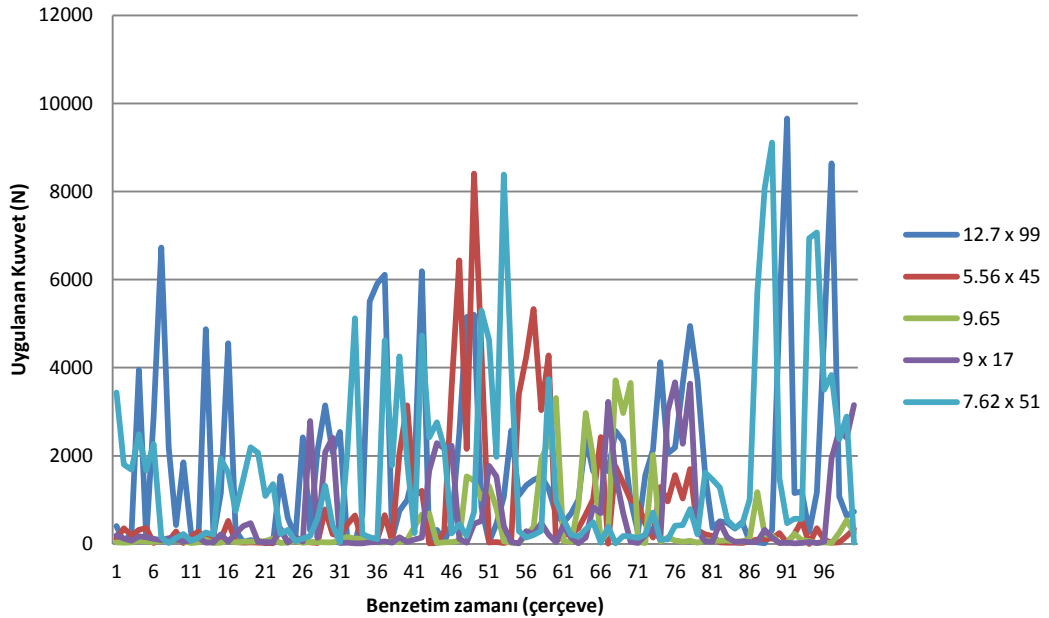
Uygulama sırasında ortaya çıkan sorunların çoğu modelleme yapılıp, gerçekleştirme kütüphanesine aktarılırken ortaya çıkmaktadır. Blender ile Ogre3D için dışarıya aktarım gerçekleştirirken kemik bilgileri de mesh dosyası içerisine aktarılmaktadır. Bir noktaya etki eden kemik sayısı en fazla dört olabilir. Eğer bu sayı dörtten fazla ise en yüksek değere sahip olan dört kemik alınır, diğerleri yok sayılır. Bunu önlemek için, dışarıya aktarım sırasında belirli bir değerin altında kalan kemiklerin yok sayılması özelliği kullanılabilir.

Eğer dışarıya aktarım yapılmadan önce yapılan döndürme, öteleme gibi işlemler modele uygulanmaz ise gerçekleştirme aşamasında hatalar ortaya çıkmaktadır. Bunu önlemek için; model dışarıya aktarılmadan önce tüm değişikliklerin modele uygulanması gereklidir.

BÖLÜM 6

BULGULAR

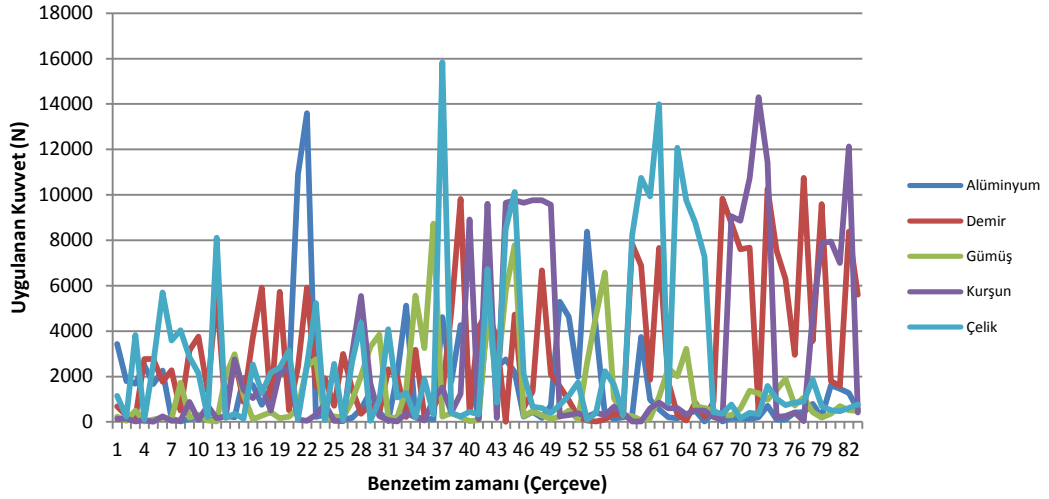
Şekil 6.1’de aynı uzaklıktan (100 m) ve aynı atış derecesiyle atılan mermilerin, alüminyum malzeme yüzeyine uyguladıkları itme kuvvetleri verilmiştir. Şekil 6.1’de görüldüğü gibi en fazla itme kuvveti uygulayan mermi tipi 12.7 mm x 9 mm’lik mermi tipidir. Daha sonra uygulanan itme kuvvetlerinin şiddeti sırasıyla; 7.62 mm x 51 mm, 5.56 mm x 45 mm, 9 mm x 17 mm ve 9.65 mm şeklindedir. Sonuç olarak yüksek başlangıç hızına ve büyük kütleye sahip olan mermilerin daha fazla itme kuvvetlerine sahip olduğu söylenebilir.



Şekil 6.1. Aynı uzaklıktan ve aynı atış derecesiyle atılan mermilerin alüminyum malzeme yüzeyine uyguladıkları itme kuvvetleri.

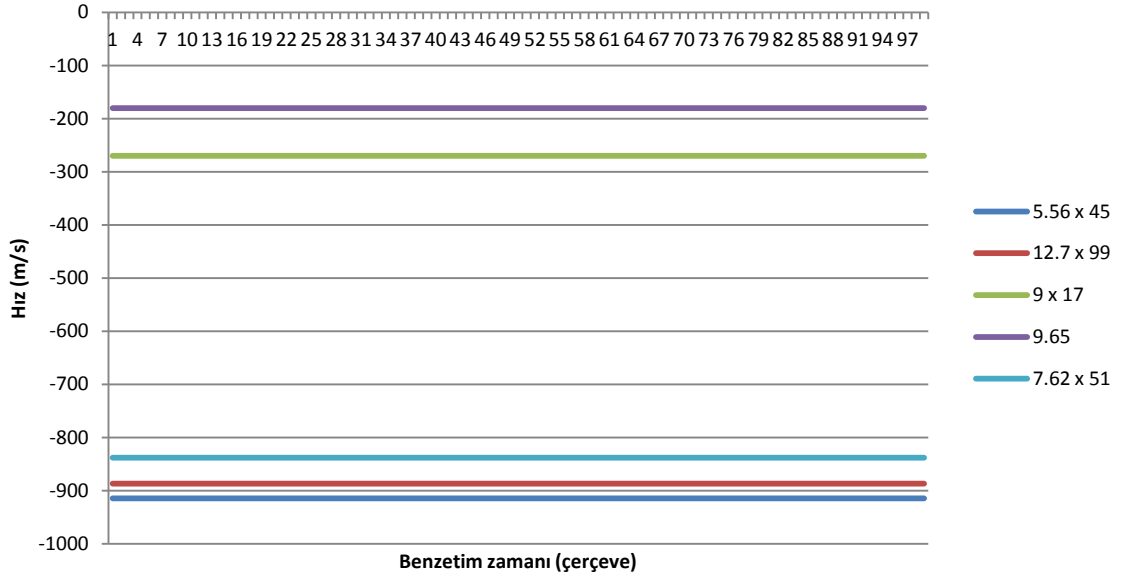
Şekil 6.2’de aynı uzaklıktan aynı atış derecesiyle atılan 7.62 mm x 51 mm mermisinin farklı malzeme yüzeylerine uyguladığı itme kuvvetleri verilmiştir. En

yüksek itme kuvveti çelik malzemesine uygulanmıştır. Daha sonra uygulanan itme sırasıyla kurşun, alüminyum, demir ve gümüştür. Bunun sebebi her bir malzemenin farklı akma mukavemetlerine sahip olması, kütlelerin farklı olmasıdır.



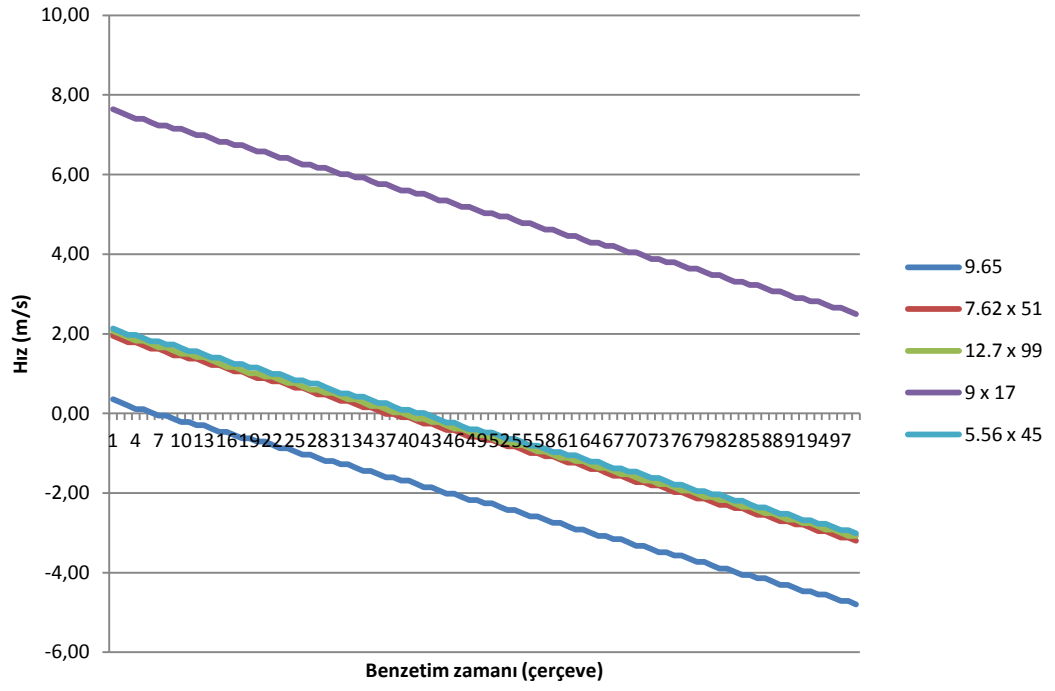
Şekil 6.2. Aynı uzaklıktan aynı atış derecesiyle atılan 7.62 mm x 51 mm mermisinin farklı malzeme yüzeylerine uyguladığı itme kuvvetleri.

Şekil 6.3'de ise aynı uzaklıktan atılan mermilerin hedefe ulaşincaya kadar olan z eksenindeki hızları verilmiştir. Mermiler üç ekseninde de hız değerine sahiptir. Fakat x eksenindeki hızları çok düşüktür. Bunun sebebi karakterin hedefe olan uzaklığıdır. Atış yapılırken, hedef ile karakter bir eksen üzerinde konumlanmıştır. Aralarındaki mesafe x ve y ekseninde çok küçük, z ekseninde ise 500 metredir. Bu yüzden z ve y eksenleri incelenebilir. x ekseninde konum değiştirme çok küçük olduğu için hızları 0 olacaktır. z eksenindeki hızları incelendiğinde, mermilerin üzerinde hiçbir kuvvet uygulanmadığı ve bu yüzden hızların sabit kaldığı görülmektedir. Hız değerleri vektörel olduğundan negatif değer alabilmektedir.



Şekil 6.3. Aynı uzaklıktan atılan mermilerin hedefe ulaşincaya kadar olan bir eksenindeki hızları.

Şekil 6.4'te aynı uzaklıktan atılan mermilerin, hedefe ulaşincaya kadar olan y eksenindeki hızları verilmiştir. y eksenindeki hızları incelendiğinde, hızların zamanla azaldığı görülmektedir. Bunun sebebi mermilere yer çekimi kuvvetinin etki etmesidir. Hedef ve karakter arasında, y eksenini üzerinde çok küçük bir mesafe farkı olduğu için bu eksen üzerindeki hızları da düşük olacaktır. Başlangıçta mermilerin y eksenindeki hızları pozitif olması, kullanılan silahın 0° 'den daha büyük bir açıyla tutulduğunu göstermektedir. Eğer silah tam 0° ile tutulursa; merminin başlangıçta y eksenindeki hızı da 0 olacaktır. Bunun sebebi y eksenini üzerinde konum değiştirmemesi olarak gösterilebilir.



Şekil 6.4. Aynı uzaklıktan atılan mermilerin hedefe ulaşmaya kadar olan y eksenindeki hızları.

BÖLÜM 7

SONUÇLAR

Bu çalışmada; farklı mermi tiplerinin, farklı malzeme yüzeylerine verdiği zararları modelleyen, işletim sisteminden bağımsız, geliştirmeye açık bir uygulama geliştirilmiştir. Uygulamaya yeni mermi tipleri ve malzeme yüzeyleri parametreleri bilindiği takdirde eklenebilmektedir.

MHS (Mermi Hasar Simülasyonu) geliştirilirken kullanılan tüm kütüphaneler, platformdan bağımsız olarak çalışan ve ücretsiz kütüphanelerden seçilmiştir. Bu sayede yapılan uygulama da platformdan bağımsız olarak çalışmaktadır.

Uygulamada kullanılan modeller Blender isimli 3B modelleme programı kullanılarak oluşturulmuştur. Sahnede yer alan tüm nesnelere bu program sayesinde modellenmiştir. Modellerin ekranda gösterilmesi için Ogre3D, benzetim işleminin gerçekleştirilmesi için ise Bullet Physics isimli kütüphane kullanılmıştır. Ara yüz oluşturmak için CEGUI isimli kütüphane, elde edilen verilerin dosya olarak kaydedilmesi için ise LibXL isimli kütüphane kullanılmıştır.

MHS tamamen ücretsiz ve açık kaynak kodlu kütüphane ve uygulamalar kullanılarak hazırlanmıştır. MHS, C++ tabanlı, işletim sisteminden bağımsız, taşınabilir, geliştirmeye açık bir uygulamadır. MHS'de kullanılan malzeme dışında kolayca farklı tipteki malzeme ve mühimmat bilgileri sisteme eklenebilir (malzeme ekleme ara yüzü) veriler dinamik olup, sonradan veri eklenip çıkarılması mümkündür.

MHS, üç temel adımda oluşturulmuştur. Birinci adımda mermi, karakter, hedef tahtası, hedef yüzey ve voksel modelleri Blender kullanarak hazırlanmıştır. İkinci adımda modeller oluşturulmuş, üçüncü adımda ise benzetim gerçekleştirilmiştir. Ayrıca bu çalışmada üç boyutlu bir benzetim uygulaması geliştirilirken izlenen

adımlar açıklanmış olup, her bir adımda uygulanacak işlemler detaylı olarak anlatılmıştır.

Benzetim sonucu elde edilen veriler grafikselleştirilerek değerlendirilmiştir. Birbirinden farklı mekanik özelliklere sahip olan malzemelerin, seçilen mermilere karşı olan davranışları grafiksel olarak gösterilmiştir. Akma mukavemeti yüksek olan malzeme yüzeylerinde delinme/parçalanma daha az olduğu belirlenmiştir. Başlangıç hızı ve kütlesi daha fazla olan mermilerin de malzeme yüzeylerine daha fazla bir itme kuvveti uyguladığı tespit edilmiştir.

Çalışmanın geleceğinde, sistemin, insanın temel hareketlerini algılama kabiliyetine sahip bir Microsoft ürünü olan kinect sensör kullanılarak bir sanal gerçeklik uygulaması hazırlanması planlanmaktadır. Bu bağlamda geliştirilen uygulama limit anahtarlar, farklı tipteki sensörler, gibi elektronik elemanlar ile zenginleştirilerek, kullanıcı etkileşiminin gerçeğe daha yakın olarak gerçekleşmesi hedeflenmektedir.

KAYNAKLAR

1. Moore, G., "Cramming more components onto integrated circuits", *Electronics Magazine*, 38: 8 (1965).
2. Angel E., and Shreiner D., "Interactive Computer Graphics A Top Down Approach With Shader Based OpenGL 6th Ed.", *Addison-Wesley*, ABD, 350-352 (2012).
3. Buss, R. B., "3-D Computer Graphics", *Cambridge University Press*, New York, 1-15 (2003).
4. Erdoğan E., Özkurt A., ve Özgören M., " Üç boyutlu etkileşimli sanal anatomi laboratuvarı tasarısı ve uygulaması", *Biomedical Engineering Meeting (BIYOMUT)*, 1-4 (2010).
5. Shirley, P., Ashikhmin, M., Gleicher, M., and Marschner, R. S., "Fundamentals of Computer Graphics", *A K Peters/CRC Press*, ABD , 1-5 (2009).
6. Çetin, A., "Bilgisayar Grafikleri", *Seçkin Yayıncılık*, Ankara, 83-103 (2003).
7. İnternet: Blender, "About Blender", <http://www.blender.org/about/> (2014).
8. İnternet: Blender, "Blender Features", <http://www.blender.org/features> (2014).
9. İnternet: Autodesk, "3D Animasyon Yazılımı, Bilgisayar Destekli Tasarım Yazılımı Maya", <http://www.autodesk.com.tr/products/autodesk-maya/overview> (2014).
10. İnternet: Autodesk, "3D modeling, animation, and rendering software | 3Ds Max", <http://www.autodesk.com/products/autodesk-3ds-max/overview> (2014).
11. Derakhshani, D., "Introducing Autodesk® Maya® 2014", *John Wiley & Sons*, Indianapolis, 2-13 (2013).
12. Lengyel E., "Mathematics for 3D Game Programming and Computer Graphics", *Course Technology*, China, 1-12 (2012).
13. İnternet: OpenGL, "OpenGL API Documentation", <http://www.opengl.org/documentation/> (2014).
14. İnternet: Microsoft, "What is DirectX", https://www.microsoft.com/resources/documentation/windows/xp/all/proddocs/en-us/what_is_directx.msp?mfr=true (2014).

15. İnternet: Ogre3D, “Getting Started”, <http://www.ogre3d.org/tikiwiki/tiki-index.php?page=Getting+Started> (2014).
16. İnternet: Ogre3D, “Addons”, <http://www.ogre3d.org/developers/addons> (2014).
17. İnternet: OpenSceneGraph, “Features”, <http://www.openscenegraph.org/index.php/about/features> (2014).
18. Mıdık, Ö., ve Kartal, M. “Simülasyona Dayalı Tıp Eğitimi”, *Marmara Medical Journal*, 23 (3): 389-399 (2010).
19. Kılıç, S. A., Sözen, M. A., Popescu, V., ve Hoffmann, C., “Bina-uçak çarpması probleminin bilgisayar ortamında simülasyonu”, *17. TMMOB Teknik Kongre ve Sergisi*, İstanbul (2004).
20. Otaduy, M. A., Garre, C., Gascón, J., Miguel, E., Pérez, Á. G., and Zurdo, J. S., “Modeling and simulation of a human shoulder for interactive medical applications”, *Proc. Spanish Conf. Comput. Graph*, İspanya, 229-237 (2010).
21. Yao, C., Xu, M., and Sui, X., “The Application of the OGRE in flight simulator visual system”, *Computer Programming Skills & Maintenance*, 12: 045 (2010).
22. Mevlütoğlu, A., “Modelleme ve simülasyon teknolojilerinin tedarik süreç yönetiminde kullanılması ve simülasyon tabanlı tedarik yöntemi”, *Savunma Sanayi Gündemi*, 11: 23-26 (2010).
23. İnternet: Coumans, E., “Bullet 2.80 Physics SDK Manual”, http://bulletphysics.com/ftp/pub/test/physics/Bullet_User_Manual.pdf (2014).
24. İnternet: Newton Game Dynamics, “Newton Game Dynamics”, <http://newtondynamics.com/forum/newton.php> (2014).
25. İnternet: Open Dynamics Engine, “Manual:Introduction”, http://ode-wiki.org/wiki/index.php?title=Manual:_Introduction (2014).
26. İnternet: Nvidia, “Nvidia Physx”, <http://www.nvidia.com.tr/object/nvidia-physx-tr.html> (2014).
27. Carlucci, E. D., and Jaconson, S. S., “Ballistics Theory and Design of Guns and Ammunition”, *CRC Press*, New York, (2007).
28. Üner, H. B. Ve Koç, S., “Ateşli Silahlar ve Balistiğin Tarihi”, Adli Tıp Kurumu, *İstanbul Cerrahpaşa Tıp Fakültesi Adli Tıp Anabilim Dalı*, İstanbul, 115-151 (1997).

29. Deniz, H., “Çift fazlı bir çeliğin 7.62 mm’lik zırh delici mermi karşısında balistik davranışının incelenmesi”, Yüksek Lisans Tezi, **Gazi Üniversitesi Fen Bilimleri Enstitüsü**, Ankara, 3-10 (2009).
30. Alim Y., “Aynı marka farklı seri numaralı tabancalarda çıkan mermi çekirdeğinin hangi tabancaya ait olduğunun saptanması için yapılan balistik analizler”, Yüksek Lisans Tezi, **Marmara Üniversitesi Fen Bilimleri Enstitüsü**, İstanbul, 8-18 (2009).
31. Özbek, T., “Mukavemet”, **Birsen Yayınevi**, İstanbul, 1-10 (1978).
32. Hibbeler, R. C., “Mühendislik Mekaniği Dinamik”, Çeviri: Soyuçok A., Soyuçok Ö., **Literatür Yayıncılık**, İstanbul, 283-300, 357-400, 413-433, 445-470 (2006).
33. Serway, A. R., ve Beichner, J. R., “Fen ve Mühendislik İçin Fizik” Çev: Çolakoğlu K., **Palme Yayıncılık**, Ankara, 23-50 (2007).
34. Aköz, Y. A., ve Omurtag, H. M., “Mühendisler İçin Mekanik Dinamik”, **Beta Basım Yayım**, İstanbul, 189-200 (1993).
35. Uçun, İ., Aslantaş, K., ve Apaydın, D., "Çok kaplamalı kesici takımla tornalama işleminin sonlu elemanlar yöntemiyle modellenmesi", **Makine Teknolojileri Elektronik Dergisi**, 7 (1): 69-82 (2010)
36. Palmer, W. B., and Oxley, P. L. B., “Mechanics of metal cutting,” **Proc.Inst. Mech. Eng.**, 173: 623–654 (1959).
37. Armstrong, R. W., and Zerilli, F. J., “Dislocation mechanics based analysis of material dynamics behavior”, **Journal De Physique**, 49 (C3): 529-534 (1988).
38. Johnson, G. R., and Cook, W. H., “A constitutive model and data for metals subjected to large strains, high strain rates and high”, **Proceedings of the 7th International Symposium on Ballistics**, Netherlands, 541–547 (2009).
39. Şenel, F., Balya, B. ve Parnas, L., “İleri kompozit zırh malzemelerin balistik analizi”, **Savunma Teknolojileri Kongresi**, Orta Doğu Teknik Üniversitesi, Ankara, 1-10 (2004).
40. Borvik, T., Olovsson, L., Dey, S., and Langseth, M., “Normal and oblique impact of small arms bullets on AA6082-T4 aluminium protective plates”, **International Journal of Impact Engineering**, 38 (7): 577-589 (2011).
41. Borvik, T., Dey, S., and Clause, A. H., “Perforation resistance of five different high-strength steel plates subjected to small-arms projectiles”, **International Journal of Impact Engineering**, 36 (7): 948-964 (2009).

42. Hub, J., “Numerical simulations of the 12.7-PZ32 bullet penetration of the aluminium alloy sheet metal”, *Academic and Applied Research in Military Science*, 8 (2): 339-344 (2009).
43. Hub, J., Komenda, J., and Novak, M., “Ballistic limit evaluation for impact of pistol projectile 9mm luger on aircraft skin metal plate”, *Advances in Military Technology*, 7 (1): 21-29 (2012).
44. Özşahin, E., ve Tolun, S., “Alüminyum levhaların yüksek hızlı çarpma davranışları için ampirik bir model”, *Havacılık ve Uzay Teknolojileri Dergisi*, 4 (2): 59-65 (2009).
45. Pechoucek, P., Rolc, S., and Buchar, J., “Fragment simulating projectile penetration into layered targets”, *Engineering Mechanics*, 18 (5,6): 353-361 (2011).
46. Mazarak, O., Martins, C., and Amanatides, J. “Animating exploding objects”, *Graphic Interface*, 211-218 (1999).
47. Terzopoulos, D., and Witkin, A., "Physically Based models with rigid and deformable components", *IEEE Computer Graphics and Applications*, 8 (6): 41-51 (1988).
48. Nealen, A., Müller, M., and Richard, K., "Physically based deformable models in computer graphics", *Eurographics* (2005).
49. Kaufman, A. E., "Volume visualization: Principles and advances", *Annual Conference on Computer Graphics – SIGGRAPH*, USA (2003).
50. Terzopoulos, D., and Fleischer, K., "Modeling inelastic deformation: viscoelasticity, plasticity, fracture", *Computer Graphics*, 22 (4): 269-278 (1988).
51. Terzopoulos, D., and Metaxas, D., "Dynamic deformation of solid primitives with constraints", *SIGGRAPH*, 26 (2): 309-312 (1992)
52. İnternet: Makina ve Kimya Endüstrisi Kurumu, “Makina ve Kimya Endüstrisi Kurumu”, <http://www.mkek.gov.tr/tr/Urunler.aspx> (2014).
53. Balentine, L., “Six Mountains (Back Office) Kindle Ed.”, *Amazon Digital Services, Inc.* (2013).

ÖZGEÇMİŞ

Rafet Durgut 1989 yılında Afyon'da doğdu; ilk ve orta öğrenimini İzmir'de tamamladı. Mimar Sinan Teknik Lisesi Endüstriyel Elektronik Bölümü'nden mezun oldu. 2007 yılında Ege Üniversitesi Ege Meslek Yüksekokulunda yükseköğrenime başlayıp 2009 yılında mezun oldu. Aynı yıl Süleyman Demirel Üniversitesi, Bilgisayar ve Kontrol Öğretmenliği programına başladı. 2012 yılında mezun oldu. Aynı yıl, araştırma görevlisi olarak Karabük Üniversitesi Bilgisayar Mühendisliği bölümüne atandı. Halen bu görevine devam etmektedir.

ADRES BİLGİLERİ

Adres : Karabük Üniversitesi
Mühendislik Fakültesi
Bilgisayar Mühendisliği Bölümü
Balıklarkayası Mevkii / KARABÜK

Tel : (553) 316 22 70
E-posta : rafetdurgut@karabuk.edu.tr