Nesneye Dayalı Programlama DERSİ

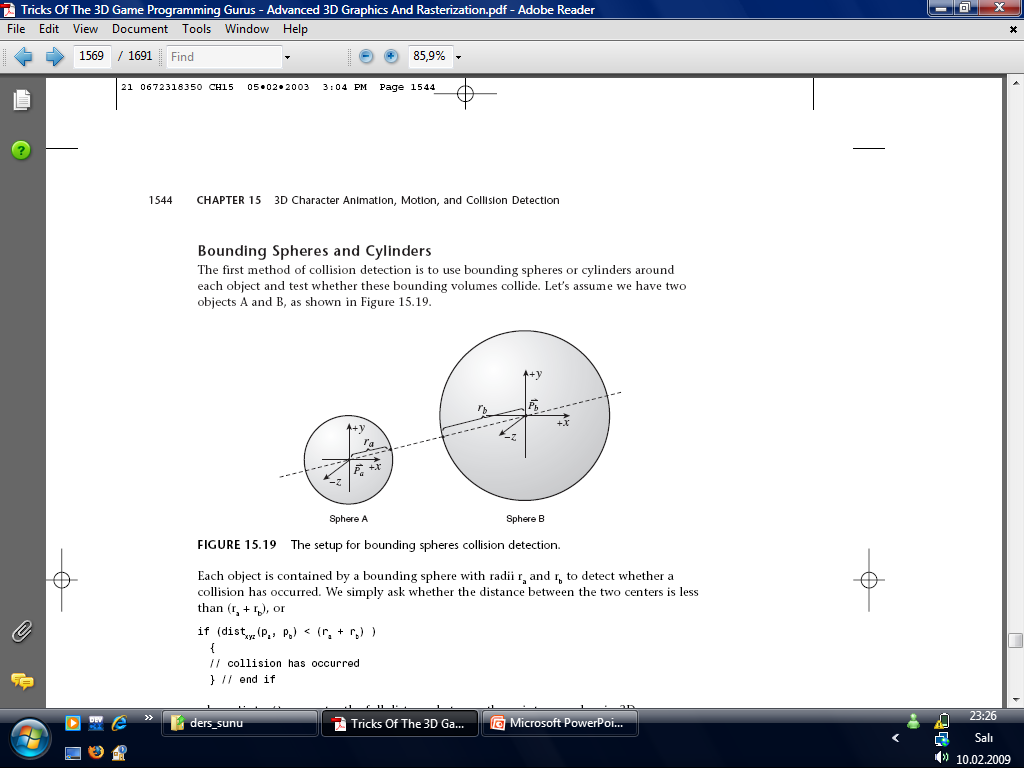
ÖDEVİ

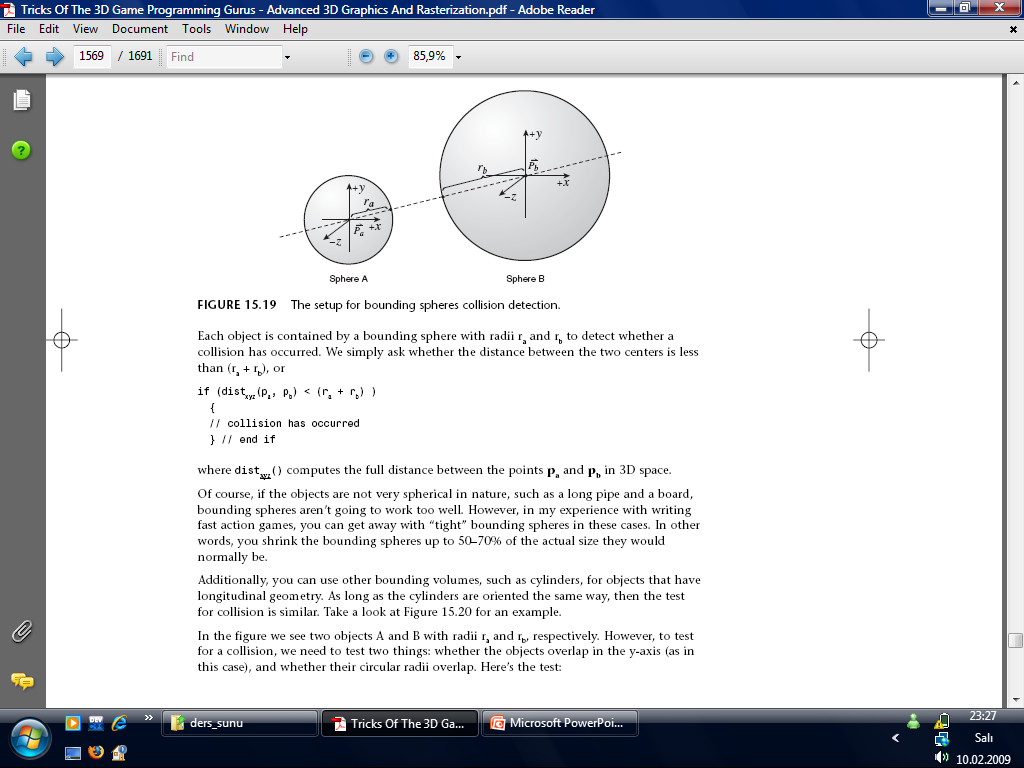
**Soru 1:**

**Bir statik sınıf oluşturarak, sınıf için aşağıdaki görevleri yerine getiren metotları gerçekleştiriniz?**

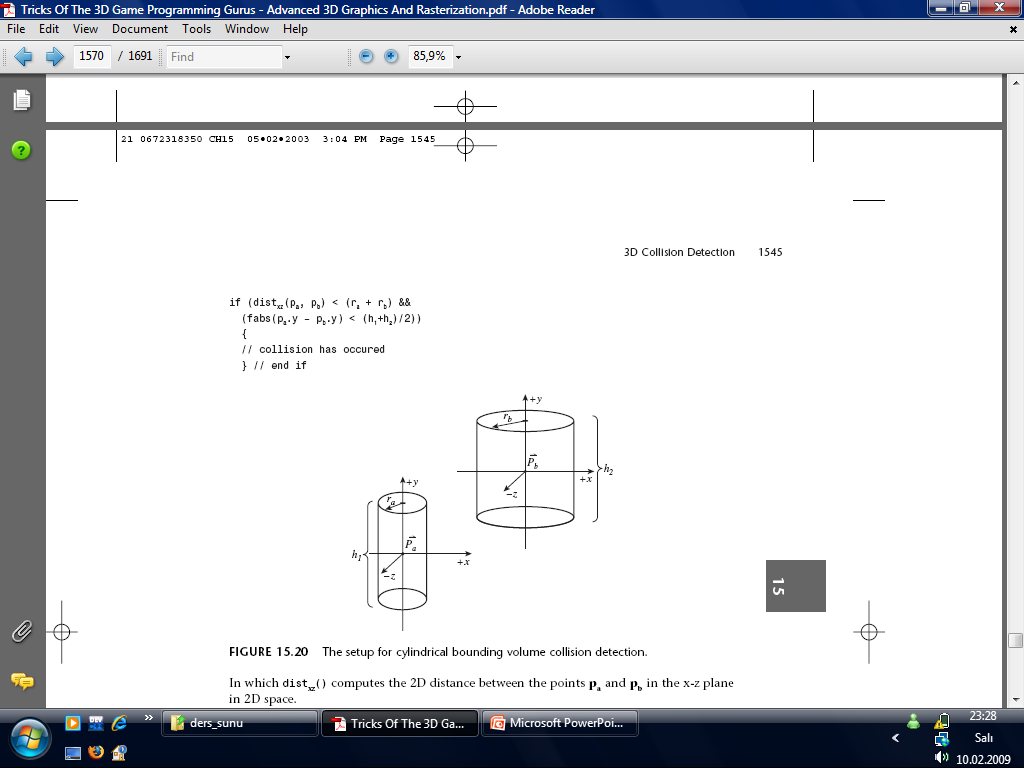
1. **Nokta, Dörtgen çarpışma denetimi**
2. **Nokta, çember çarpışma denetimi**
3. **Dikdörtgen, dikdörtgen çarpışma denetimi**
4. **Dikdörtgen, çember çarpışma denetimi**
5. **Çember, çember çarpışma denetimi**
6. **Nokta, Küre çarpışma denetimi**
7. **Nokta, dikdörtgen prizma çarpışma denetimi**
8. **Nokta, Silindir çarpışma denetimi**
9. **Silindir, silindir çarpışma denetimi**
10. **Küre, küre çarpışma denetimi**
11. **Küre silindir çarpışma denetimi**
12. **Yüzey, küre çarpışma denetimi**
13. **Yüzey, dikdörtgen prizma çarpışma denetimi**
14. **Yüzey silindir çarpışma denetimi**
15. **Küre, dikdörtgen prizma çarpışma denetimi**
16. **Dikdörtgen prizma, dikdörtgen prizma çarpışma denetimi**

**Soru-2 Bir grafik arayüzü oluşturunuz(Windows Form uygulaması)? Daha sonra genelleyici sınıf nesnesi oluşturunuz(List, Dictionary vb)? Bu nesneye çok sayıda geometrik nesneyi(8 adet) ekleyiniz. Eklenen nesneleri çizdiriniz? Sonra bu nenelerin serbest hareketleri için çarpışma denetimi gerçekleştiriniz?**

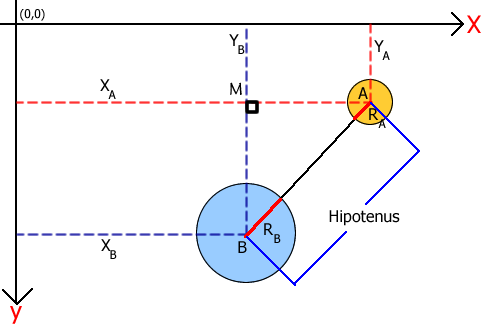
****

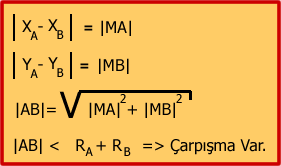
****

**Silindir silindir çarpışması**

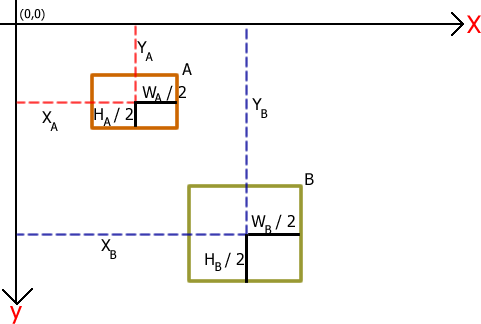
****

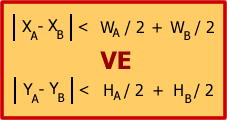
**Çember çember çarpışması**

****

****

**Dikdörtgen, dikdörtgen çarpışması**

****

****

**Bazı Kod Örnekleri**

Dikdörtgen.cs

|  |  |
| --- | --- |
|  | using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace ConsoleApp1  {  public class dikdortgen  {  point m; int en;int boy;  public dikdortgen()  {  M = new point(); //köşe noktası  En = 0;Boy = 0;  }  public dikdortgen(point p, int en,int boy)  { M = p; En =en; Boy = boy; }  public point M { get => m; set => m = value; }  public int En { get => en; set => en = value; }  public int Boy { get => boy; set => boy = value; }  }  } |

Cylender.cs

|  |  |
| --- | --- |
|  | using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace ConsoleApp1  {  public class cylender  {  point3d m; int r; int h;  public cylender()  {  M = new point3d();// merkez noktası  R = 0;  H = 0;  }  public cylender(point3d p, int r, int h)  { M = p; R = r;H = h; }  public int R { get => r; set => r = value; }  public int H { get => h; set => h = value; }  internal point3d M { get => m; set => m = value; }  }  } |

Kure.cs

|  |  |
| --- | --- |
|  | using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace ConsoleApp1  {  public class kure  {  point3d m; int r;  public kure()  {  M = new point3d();// merkez noktası  R = 0;  }  public kure(point3d p, int r)  { M = p; R = r; }  public int R { get => r; set => r = value; }  internal point3d M { get => m; set => m = value; }  }  } |

Program.cs (class point, point3d, circle, carpisma ve program yer almakta)

|  |  |
| --- | --- |
|  | using System;  namespace ConsoleApp1  {  public class point  {  int x;  int y;  public point()  { X = 0;Y = 0; }  public point(int x,int y)  { X = x;Y = y; }  public int X { get => x; set => x = value; }  public int Y { get => y; set => y = value; }  }  public class point3d : point  {  int z;  public point3d():base()  { Z = 0; }  public point3d(int x, int y,int z)  { Z = z; }  public int Z { get => z; set => z = value; }  }  public class circle  {  point m; int r;  public circle()  {  M = new point();// merkez noktası  R = 0;  }  public circle(point p,int r)  { M = p;R = r; }    public int R { get => r; set => r = value; }  internal point M { get => m; set => m = value; }  }  public static class Carpisma  {  public static bool circleCarp(circle c1, circle c2)  {  float d=(float)Math.Sqrt(Math.Pow((c1.M.X - c2.M.X), 2) + Math.Pow((c1.M.Y - c2.M.Y), 2));  //Console.WriteLine(d);  if ((c1.R + c2.R) >d)  return true ;  else  return false;  }  public static bool kureCarp(kure k1, kure k2)  {  float d = (float)Math.Sqrt(Math.Pow((k1.M.X - k2.M.X), 2) +  Math.Pow((k1.M.Y - k2.M.Y), 2)+ Math.Pow((k1.M.Z - k2.M.Z), 2));  //Console.WriteLine(d);  if ((k1.R + k2.R) >(int)d)  return true;  else  return false;  }  public static bool dikdortgenCarp(dikdortgen d1, dikdortgen d2)  {  int Xa = d1.M.X + d1.En / 2;  int Ya = d1.M.Y + d1.Boy / 2;  int Xb = d2.M.X + d2.En / 2;  int Yb = d2.M.Y + d2.Boy / 2;  if ( Math.Abs(Xa-Xb) <(d1.En/2+d2.En/2) && Math.Abs(Ya - Yb) < (d1.Boy /2 + d2.Boy / 2))  return true;  else  return false;  }  public static bool cylenderCarp(cylender k1, cylender k2)  {  point3d pa = new point3d(k1.M.X, k1.M.Y + k1.H / 2, k1.M.Z);  point3d pb = new point3d(k2.M.X, k2.M.Y + k2.H / 2, k2.M.Z);  float d= (float)Math.Sqrt(Math.Pow((pa.X - pb.X), 2) +  Math.Pow((pa.Y - pb.Y), 2) + Math.Pow((pa.Z - pb.Z), 2));  //Console.WriteLine(d);  if ((k1.R + k2.R) > (int)d && Math.Abs(pa.Y-pb.Y)<((k1.H+k2.H)/2))  return true;  else  return false;  }  }  class Program  {  static void Main(string[] args)  {  circle c1 = new circle(new point(5, 5), 3);  circle c2 = new circle(new point(15, 15), 3);  if(Carpisma.circleCarp(c1, c2))  Console.WriteLine(" daireler çarpışıyor");  else  Console.WriteLine(" daireler çarpışmıyor");  if (Carpisma.dikdortgenCarp(new dikdortgen(new point(5,5), 10,10), new dikdortgen(new point(20, 30), 10, 10)))  Console.WriteLine(" dikdörtgenler çarpışıyor");  else  Console.WriteLine(" dikdörtgenler çarpışmıyor");  if (Carpisma.kureCarp(new kure(new point3d(5, 5, 5), 10), new kure(new point3d(100, 100, 100), 5)))  Console.WriteLine(" küreler çarpışıyor");  else  Console.WriteLine(" küreler çarpışmıyor");  if (Carpisma.cylenderCarp(new cylender(new point3d(5, 5, 5), 6,10), new cylender(new point3d(20, 40, 10), 5,12)))  Console.WriteLine(" silindirler çarpışıyor");  else  Console.WriteLine(" silindirler çarpışmıyor");  }  }  } |

**İstenenler:**

1. **Proje için gerekli işlemleri yapan C# kaynak kod dosyası**
2. **Projede yapılan işlemlerin basit bir şekilde anlatıldığı rapor dosyası**

# Ödevin son teslim tarihi : sistemde verilen süre ve tarihtir.

UYULMASI GEREKEN KOD DÜZENİ

Her C++ dosyasının başında aşağıdaki yorum bloğu bulunacaktır. Yorum bulunmayan her C++ dosyası için 10 puan kırılacaktır. (**pdf üzerinden kopyalayıp yapıştırmanız problem çıkartabilir**)

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\* SAKARYA ÜNİVERSİTESİ

\*\* BİLGİSAYAR VE BİLİŞİM BİLİMLERİ FAKÜLTESİ

\*\* BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

\*\* PROGRAMLAMAYA GİRİŞİ DERSİ

\*\*

\*\* ÖDEV NUMARASI…...:

\*\* ÖĞRENCİ ADI...............:

\*\* ÖĞRENCİ NUMARASI.:

\*\* DERS GRUBU…………:

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

Değişken isimleri anlamlı olmalıdır. Örneğin tek sayı adedini tutacak bir değişken için

|  |  |  |  |
| --- | --- | --- | --- |
| YANLIŞ | yerine | DOĞRU | yazılmalıdır. |
| int a; | int tekAdedi; |

Her bir küme parantezinin altındaki kodları yazmadan önce **tab** tuşuna basılarak boşluk bırakılmalıdır.

|  |  |  |
| --- | --- | --- |
| YANLIŞ |  | DOĞRU |
| if(true)  {  int a =6;  cout<<"Merhaba";  if(false)  cout<<"Nasilsin";  } |  | if(true)  {  int a =6;  cout<<"Merhaba";  if(false)  cout<<"Nasilsin";  } |

Her döngü ve koşul işlemlerinden önce yapılan işlem hakkında yorum yazılmalıdır

|  |  |  |
| --- | --- | --- |
| YANLIŞ |  | DOĞRU |
| int t = 0;  for(int i=0;i<100;i++)  t+=i; |  | int toplam = 0;  //1 den 100 e kadar olan sayılar toplanıyor  for(int i=0;i<100;i++)  toplam+=i; |

ÖDEV TESLİM KURALLARI

# ELEKTRONİK olarak ödev yükleme sistemi İLE TESLİM EDİLECEK DOKÜMANLAR

* Ödev için bir rapor dosyası hazırlanacaktır.
* Bu dosyanın ilk sayfasında kapak sayfası bulunacaktır.
* Dosya ‘pdf’ formatına çevrilip yollanması gerekmektedir.

# DERLEYİCİ VE TESLİM TARİHİ

Derleme için **Visual Studio 2017 ve üzeri** C# derleyicisi kullanılacaktır.

**Zamanında teslim edilmeyen ödevler değerlendirilmeyecektir.**

**Teslim saati ve günü sistemde belirtilmiştir**

KOD PAYLAŞMAK YASAKTIR

Ödevler bireyseldir ve verilen bütün ödevler karşılaştırılacaktır. Birbirine çok benzeyen ödevler **kopya** muamelesi görecektir. Öğretim üyesi kopya durumunda ödevi değerlendirmez veya gerekli soruşturma ve ceza işlemlerini başlatabilir.

**Önemli NOT:** Sistemde size verilen üç defa yükleme hakkı üç farklı dosya için değildir. Sadece son yüklediğiniz dosya sistemde tutulmaktadır. Ödevinizi tamamanı bitirdikten sonra ve tamamını yükleyiniz.