[Tarih]

Yiğit Karakoyun

Tetris Oyunu Komutları

|  |
| --- |
| GOSTER CLASS |
| using System;  using System.Collections.Generic;  using System.Windows.Forms;  namespace GoodDuck  {  class GOSTER  {  List<PictureBox> resimler;  public GOSTER()  {  this.resimler = legoHepsi.resimler;  }    //SONRA YENİDEN GÖSTER  public void Goster()  {  if (legoHepsi.lego1 != null)  {  Yapi temp = legoHepsi.lego1.Simdiki;  try  {  if (resimler != null)  {  resimler[temp.a].BackgroundImage = legoHepsi.lego1.resim;  resimler[temp.b].BackgroundImage = legoHepsi.lego1.resim;  resimler[temp.c].BackgroundImage = legoHepsi.lego1.resim;  resimler[temp.d].BackgroundImage = legoHepsi.lego1.resim;  }  else  {  MessageBox.Show("Boş resimler");  }    }  catch (Exception)  {  MessageBox.Show("Boya gösterilemedi");  }    }  }  //ÖNCE TEMİZLE  public void Temizle()  {  if (legoHepsi.lego1 != null)  {  Yapi temp = legoHepsi.lego1.Simdiki;  try  {  if (legoHepsi.lego1.hareketeDevam)  {  resimler[temp.a].BackgroundImage = Resimler.varsayilan;  resimler[temp.b].BackgroundImage = Resimler.varsayilan;  resimler[temp.c].BackgroundImage = Resimler.varsayilan;  resimler[temp.d].BackgroundImage = Resimler.varsayilan;  }    }  catch (Exception)  {  MessageBox.Show("Boya TemizleNemedi");  }  }  }  }  } |
|  |

|  |
| --- |
| HIT CLASS |
| using System.Collections.Generic;  namespace GoodDuck  {  class HIT  {  LEGOYAP legoMaker;    public HIT(LEGOYAP legoMaker)  {  this.legoMaker = legoMaker;  }  #region SOL DUVAR && NESNE  public void solDuvar()  {  legoHit.solDuvar = true;  LegoOrtak lego1 = legoHepsi.lego1;  if ((lego1.Simdiki.a - 1) % 10 == 9)  {  legoHit.solDuvar = false;  }  if ((lego1.Simdiki.b - 1) % 10 == 9)  {  legoHit.solDuvar = false;  }  if ((lego1.Simdiki.c - 1) % 10 == 9)  {  legoHit.solDuvar = false;  }  if ((lego1.Simdiki.d - 1) % 10 == 9)  {  legoHit.solDuvar = false;  }  }  public void solNesne()  {  LegoOrtak lego1 = legoHepsi.lego1;  List<int> legolar = legoHepsi.legolar;  legoHit.solNesne = true;  if (legolar.Contains(lego1.Simdiki.a - 1))  {  legoHit.solNesne = false;  }  if (legolar.Contains(lego1.Simdiki.b - 1))  {  legoHit.solNesne = false;  }  if (legolar.Contains(lego1.Simdiki.c - 1))  {  legoHit.solNesne = false;  }  if (legolar.Contains(lego1.Simdiki.d - 1))  {  legoHit.solNesne = false;  }    }  public string solla()  {  LegoOrtak lego1 = legoHepsi.lego1;  solNesne();  solDuvar();  if (legoHit.solDuvar && legoHit.solNesne)  {  lego1.Hareketli--;  }  return(lego1.Adi + " solla " + lego1.Simdiki.a + " " + lego1.Simdiki.b + " " + lego1.Simdiki.c + " " + lego1.Simdiki.d);  }  #endregion  #region SAĞ DUVAR && NESNE  public void sagDuvar()  {  LegoOrtak lego1 = legoHepsi.lego1;  legoHit.sagDuvar = true;  if ((lego1.Simdiki.a + 1) % 10 == 0)  {  legoHit.sagDuvar = false;  }  if ((lego1.Simdiki.b + 1) % 10 == 0)  {  legoHit.sagDuvar = false;  }  if ((lego1.Simdiki.c + 1) % 10 == 0)  {  legoHit.sagDuvar = false;  }  if ((lego1.Simdiki.d + 1) % 10 == 0)  {  legoHit.sagDuvar = false;  }  }  public void sagNesne()  {  LegoOrtak lego1 = legoHepsi.lego1;  List<int> legolar = legoHepsi.legolar;  legoHit.sagNesne = true;  if (legolar.Contains(lego1.Simdiki.a + 1))  {  legoHit.sagNesne = false;  }  if (legolar.Contains(lego1.Simdiki.b + 1))  {  legoHit.sagNesne = false;  }  if (legolar.Contains(lego1.Simdiki.c + 1))  {  legoHit.sagNesne = false;  }  if (legolar.Contains(lego1.Simdiki.d + 1))  {  legoHit.sagNesne = false;  }  }  public string sagla()  {  LegoOrtak lego1 = legoHepsi.lego1;  sagNesne();  sagDuvar();  if (legoHit.sagDuvar && legoHit.sagNesne)  {  lego1.Hareketli++;  }  return(lego1.Adi + " sagla " + lego1.Simdiki.a + " " + lego1.Simdiki.b + " " + lego1.Simdiki.c + " " + lego1.Simdiki.d);  }  #endregion  #region AŞAĞI DUVAR && NESNE  public void asagiDuvar()  {  LegoOrtak lego1 = legoHepsi.lego1;  legoHit.asagiDuvar = true;  if ((lego1.Simdiki.a) > 189)  {  legoHit.asagiDuvar = false;  }  if ((lego1.Simdiki.b) > 189)  {  legoHit.asagiDuvar = false;  }  if ((lego1.Simdiki.c) > 189)  {  legoHit.asagiDuvar = false;  }  if ((lego1.Simdiki.d) > 189)  {  legoHit.asagiDuvar = false;  }  }  public void asagiNesne()  {  LegoOrtak lego1 = legoHepsi.lego1;  List<int> legolar = legoHepsi.legolar;  legoHit.asagiNesne = true;  if (legolar.Contains(lego1.Simdiki.a + 10))  {  legoHit.asagiNesne = false;  }  if (legolar.Contains(lego1.Simdiki.b + 10))  {  legoHit.asagiNesne = false;  }  if (legolar.Contains(lego1.Simdiki.c + 10))  {  legoHit.asagiNesne = false;  }  if (legolar.Contains(lego1.Simdiki.d + 10))  {  legoHit.asagiNesne = false;  }  }    public string asagila()  {  LegoOrtak lego1 = legoHepsi.lego1;  asagiNesne();  asagiDuvar();  if (lego1.hareketeDevam)  {  if (legoHit.asagiNesne && legoHit.asagiDuvar)  {  lego1.Hareketli += 10;  lego1.hareketeDevam = true;  }  else  {  lego1.resim = Resimler.yapisik;  lego1.hareketeDevam = false;  SKORLAR.skorHesaplaBiriktir();  }  }  else  {    }    return (lego1.Adi + " asagila " + lego1.Simdiki.a + " " + lego1.Simdiki.b + " " + lego1.Simdiki.c + " " + lego1.Simdiki.d+" "+lego1.hareketeDevam);  }  #endregion  #region -------------------------DONME SOL SAĞ AŞAĞI DUVAR && NESNE------------  public void donmeDuvarYan()  {  LegoOrtak lego1 = legoHepsi.lego1;  Yapi sonraki = lego1.Sonraki;  int a = sonraki.a % 10;  int b = sonraki.b % 10;  int c = sonraki.c % 10;  int d = sonraki.d % 10;  bool dokuz = false, sifir = false;  if (a == 0 || b == 0 || c == 0 || d == 0) sifir = true;  if (a == 9 || b == 9 || c == 9 || d == 9) dokuz = true;  legoHit.donmeDuvarYan = true;  if (dokuz && sifir)  {  legoHit.donmeDuvarYan = false;    }  }    public void donmeAsagiDuvar()  {  LegoOrtak lego1 = legoHepsi.lego1;  Yapi sonraki = lego1.Sonraki;  legoHit.donmeDuvarYer = true;  if ((sonraki.a) > 189)  {  legoHit.donmeDuvarYer = false;  }  if ((sonraki.b) > 189)  {  legoHit.donmeDuvarYer = false;  }  if ((sonraki.c) > 189)  {  legoHit.donmeDuvarYer = false;  }  if ((sonraki.d) > 189)  {  legoHit.donmeDuvarYer = false;  }  }  public void donmeNesne()  {  LegoOrtak lego1 = legoHepsi.lego1;  List<int> legolar = legoHepsi.legolar;  Yapi sonraki = lego1.Sonraki;  legoHit.donmeNesne = true;  if (legolar.Contains(sonraki.a + 10))  {  legoHit.donmeNesne = false;  }  if (legolar.Contains(sonraki.b + 10))  {  legoHit.donmeNesne = false;  }  if (legolar.Contains(sonraki.c + 10))  {  legoHit.donmeNesne = false;  }  if (legolar.Contains(sonraki.d + 10))  {  legoHit.donmeNesne = false;  }  }  public string dondurme()  {  LegoOrtak lego1 = legoHepsi.lego1;  donmeAsagiDuvar();  donmeNesne();  donmeDuvarYan();    if (lego1.hareketeDevam)  {  if (legoHit.donmeDuvarYan && legoHit.donmeDuvarYer && legoHit.donmeNesne)  {  lego1.Ilerle() ;    }  }  else  {  lego1.resim = Resimler.yapisik;  }  return (legoHit.donmeDuvarYan + "");  }  #endregion  }  } |
|  |

|  |
| --- |
| RENKLER, RESİMLER, LEGOORTAK, LEGOHEPSİ, LEGOHIT, YAPI, LEGOORTAK CLASSLARI |
| using System.Collections.Generic;  using System.Drawing;  using System.Windows.Forms;  namespace GoodDuck  {  public static class Renkler  {  public static Color varsayilan { get; set; } = Color.FromArgb(171,171,171) ;  public static Color yapisik { get; set; } = Color.White;  public static Color legoO { get; set; } = Color.CornflowerBlue;  public static Color legoI { get; set; } = Color.Blue;  public static Color legoS { get; set; } = Color.SeaGreen;  public static Color legoZ { get; set; } = Color.CadetBlue;  public static Color legoL { get; set; } = Color.Crimson;  public static Color legoJ { get; set; } = Color.LightGreen;  public static Color legoT { get; set; } = Color.Magenta;  }  public class Resimler  {  public static Image varsayilan { get { return null; } }  public static Image yapisik { get; set; }    public static Image legoO { get; private set; }  public static Image legoI { get; private set; }  public static Image legoS { get; private set; }  public static Image legoZ { get; private set; }  public static Image legoL { get; private set; }  public static Image legoJ { get; private set; }  public static Image legoT { get; private set; }  public static void resimYukle(int i)  {  if (i == 0)  {  Resimler.legoO = Image.FromFile("Tetrisler/" + "0" + ".PNG");  }  else if(i == 1)  {  Resimler.legoI = Image.FromFile("Tetrisler/" + "1" + ".PNG");  }  else if (i == 2)  {  Resimler.legoS = Image.FromFile("Tetrisler/" + "2" + ".PNG");  }  else if (i == 3)  {  Resimler.legoZ = Image.FromFile("Tetrisler/" + "3" + ".PNG");  }  else if (i == 4)  {  Resimler.legoL = Image.FromFile("Tetrisler/" + "4" + ".PNG");  }  else if (i == 5)  {  Resimler.legoJ = Image.FromFile("Tetrisler/" + "5" + ".PNG");  }  else if (i == 6)  {  Resimler.legoT = Image.FromFile("Tetrisler/" + "6" + ".PNG");  }  else  {  Resimler.yapisik = Image.FromFile("Tetrisler/" + "7" + ".PNG");  }  }  }  public class legoHepsi  {  public static LegoOrtak lego1 { get; set; }  public static LegoOrtak lego2 { get; set; }  public static List<PictureBox> resimler { get; set; }  public static List<int>legolar { get; set; }  public static bool OyunCalisiyorMu { get; set; } = true;  public static bool SesCalisiyorMu { get; set; } = true;  }  public class legoHit  {  public static bool solDuvar { get; set; }  public static bool solNesne { get; set; }  public static bool sagDuvar { get; set; }  public static bool sagNesne { get; set; }  public static bool asagiDuvar { get; set; }  public static bool asagiNesne { get; set; }  public static bool donmeDuvarYan { get; set; }  public static bool donmeDuvarYer { get; set; }  public static bool donmeNesne { get; set; }  }  public class Yapi  {  #region a b c d kareleri GÖRÜNTÜLEMEDE KULLANACAZ  public int a { get; private set; }  public int b { get; private set; }  public int c { get; private set; }  public int d { get; private set; }  public Yapi(int a1,int b1,int c1,int d1)  {  int a2,b2,c2,d2;  a2 = kareleriGercekKonumunaDonusturme(a1);  b2 = kareleriGercekKonumunaDonusturme(b1);  c2 = kareleriGercekKonumunaDonusturme(c1);  d2 = kareleriGercekKonumunaDonusturme(d1);  Yukle(a2,b2,c2,d2);  }  private int kareleriGercekKonumunaDonusturme(int sayi)  {  int temp=-1;  if (sayi < 4)  {  temp = sayi;  }  else if (sayi < 8)  {  temp = (sayi % 4) + 10;  }  else if (sayi < 12)  {  temp = (sayi % 4) + 20;  }  else  {  temp = (sayi % 4) + 30;  }  return temp;  }  public Yapi()  {  }  public void Yukle(int a1, int b1, int c1, int d1)  {  a = a1;  b = b1;  c = c1;  d = d1;  }  #endregion  }  public class LegoOrtak  {  #region RENK  public Image resim { get; set; }  #endregion  #region HAREKET LİSTESİ  public bool sola { get; set; } = true;  public bool saga { get; set; } = true;  public bool asagi { get; set; } = true;  public bool donek { get; set; } = true;  public bool hareketeDevam { get; set; } = true;  #endregion  public string Adi { get; set; }  public int No { get; set; }  public int Secili;    public List<Yapi> Dondur { get; set; } = new List<Yapi>();  public int Adeti { get; private set; } = 0;  public int Hareketli { get; set; } = 0;  public int Hareketsiz { get; private set; } = 3;  public bool EkleLegolar()  {  bool temp = false;  Yapi ortak = legoHepsi.lego1.Simdiki;  int a=ortak.a, b = ortak.b, c = ortak.c, d = ortak.d;  if (legoHepsi.legolar.Contains(a)|| legoHepsi.legolar.Contains(b) || legoHepsi.legolar.Contains(c) || legoHepsi.legolar.Contains(d))  {  temp = true;  }  else  {  legoHepsi.legolar.Add(ortak.a);  legoHepsi.legolar.Add(ortak.b);  legoHepsi.legolar.Add(ortak.c);  legoHepsi.legolar.Add(ortak.d);  legoHepsi.legolar.Sort();  }  return temp;  }  public void EkleKare(int a, int b, int c, int d)  {  #region KARE DONDURME DURUMLARINI KAYDET  Yapi temp = new Yapi(a, b, c, d);  Dondur.Add(temp);  Adeti = Dondur.Count;  #endregion    }  #region DONDURME İŞLEMLERİ İÇİN İLERLEME KOMUTLARI  #region SONRAKİ SEÇİMİ ÖĞREN  private int OgrenSonrakiSecim()  {  return (Secili + 1) % Adeti;  }  private int OgrenSimdikiSecim()  {  return (Secili) % Adeti;  }  private Yapi SimdikiYapi  {  get { return Dondur[OgrenSimdikiSecim()]; }  }  private Yapi SonrakiYapi  {  get { return Dondur[OgrenSonrakiSecim()]; }    }  private Yapi simdiki = new Yapi();  public Yapi Simdiki  {  get {  int a = 0, b = 0, c = 0, d = 0;    if (SimdikiYapi != null)  {  a = SimdikiYapi.a + Hareketli + Hareketsiz;  b = SimdikiYapi.b + Hareketli + Hareketsiz;  c = SimdikiYapi.c + Hareketli + Hareketsiz;  d = SimdikiYapi.d + Hareketli + Hareketsiz;    }  simdiki.Yukle(a, b, c, d);  return simdiki;  }  }  private Yapi sonraki = new Yapi();  public Yapi Sonraki  {  get  {  int a = 0, b = 0, c = 0, d = 0;  if (SonrakiYapi != null)  {  a = SonrakiYapi.a + Hareketli + Hareketsiz;  b = SonrakiYapi.b + Hareketli + Hareketsiz;  c = SonrakiYapi.c + Hareketli + Hareketsiz;  d = SonrakiYapi.d + Hareketli + Hareketsiz;  }  sonraki.Yukle(a, b, c, d);  return sonraki;  }  }  #endregion  #region İLERLETİP DONDURME KOMUTLARI  public void Ilerle()  {  //SEÇİLİYİ DEĞİŞİR  Secili = OgrenSonrakiSecim();  }  #endregion  #endregion    }  } |
|  |

|  |
| --- |
| LEGO ÖRNEKLEMELERİ CLASSLARI( LEGOS, LEGOZ…) |
| namespace GoodDuck  {  public class LegoO : LegoOrtak  {  public LegoO()  {  Adi = "O";  No = 0;  #region Lego Durum 1  EkleKare(5, 6, 9, 10);  resim = Resimler.legoO;  #endregion  }  }  public class LegoI : LegoOrtak  {  public LegoI()  {  Adi = "I";  No = 1;  #region Lego Durumlar  EkleKare(4, 5, 6, 7);  EkleKare(2, 6, 10, 14);  resim = Resimler.legoI;  #endregion  }  }  public class LegoS : LegoOrtak  {  public LegoS()  {  Adi = "S";  No = 2;  #region Lego Durumlar  EkleKare(6, 7, 9, 10);  EkleKare(2, 6, 7, 11);  resim = Resimler.legoS;  #endregion  }  }  public class LegoZ : LegoOrtak  {  public LegoZ()  {  Adi = "Z";  No = 3;  #region Lego Durumlar  EkleKare(5, 6, 10, 11);  EkleKare(3, 6, 7, 10);  resim = Resimler.legoZ;  #endregion  }  }  public class LegoL : LegoOrtak  {  public LegoL()  {  Adi = "L";  No = 4;  #region Lego Durumlar  EkleKare(5, 6, 7, 9);  EkleKare(2, 6, 10, 11);  EkleKare(3, 5, 6, 7);  EkleKare(1,2, 6, 10);  resim = Resimler.legoL;  #endregion  }  }  public class LegoJ : LegoOrtak  {  public LegoJ()  {  Adi = "J";  No = 5;  #region Lego Durumlar  EkleKare(5, 6, 7, 11);  EkleKare(2, 3, 6, 10);  EkleKare(1, 5, 6, 7);  EkleKare(2, 6, 9, 10);  resim = Resimler.legoJ;  #endregion  }  }  public class LegoT : LegoOrtak  {  public LegoT()  {  Adi = "T";  No = 6;  #region Lego Durumlar  EkleKare(5, 6, 7, 10);  EkleKare(2, 6, 7, 10);  EkleKare(2, 5, 6, 7);  EkleKare(2, 5, 6, 10);  resim = Resimler.legoT;  #endregion  }  }  } |
|  |

|  |
| --- |
| LEGOSİLME CLASS |
| using System.Collections.Generic;  using System.Windows.Forms;  namespace GoodDuck  {  class LEGOSILME  {  public List<PictureBox> resimler;  public List<int> legolar;    public int VarMi(int sayi)  {  int temp = -1;  if (legolar.Contains(sayi))  {  if (legolar.Contains(sayi+1))  {  if (legolar.Contains(sayi+2))  {  if (legolar.Contains(sayi+3))  {  if (legolar.Contains(sayi+4))  {  if (legolar.Contains(sayi+5))  {  if (legolar.Contains(sayi+6))  {  if (legolar.Contains(sayi+7))  {  if (legolar.Contains(sayi+8))  {  if (legolar.Contains(sayi+9))  {  temp = sayi;  }  }  }  }  }  }  }  }  }  }  return temp;  }  public void OYNAT()  {  int sayi = -1;  for (int i = 190; i > -1; i-=10)  {  sayi = VarMi(i); if (sayi != -1)  {  SatirSil(sayi); i += 10;  }  }    }  public void SatirSil(int satir)  {  SKORLAR.skorHesaplaSil();  foreach (int item in legoHepsi.legolar)  {  legoHepsi.resimler[item].BackgroundImage = Resimler.varsayilan;  }  TemizleKaydi(satir);  SatirKaydir(satir);  foreach (int item in legoHepsi.legolar)  {  legoHepsi.resimler[item].BackgroundImage = Resimler.yapisik;  }  }  private void TemizleKaydi(int indeks)  {  for (int i = indeks + 9; i >= indeks; i--)  {  if (legoHepsi.legolar.Contains(i))  {  if (i >= indeks)  {  legoHepsi.legolar.Remove(i);  i++;  }  else break;  }  }  }  private void SatirKaydir(int satir)  {  for (int i = 0; i < legoHepsi.legolar.Count; i++)  {  int item = legoHepsi.legolar[i];  if (item < satir)  {  legoHepsi.legolar[i] += 10;  }  }  }  public void Temizle(int indeks)  {  resimler[indeks].BackgroundImage = Resimler.varsayilan;  }  public void Tasima(int indeks)  {  resimler[indeks].BackgroundImage = Resimler.yapisik;  }    }  } |
|  |

|  |
| --- |
| LEGOYAP CLASS |
| using System;  namespace GoodDuck  {  class LEGOYAP  {  public LEGOYAP()  {  legoHepsi.lego1 = LegoYap();  legoHepsi.lego2 = LegoYap();  }  public void Sonraki(bool temp = true)  {  legoHepsi.lego1 = null;  legoHepsi.lego1 = legoHepsi.lego2;  if (temp) legoHepsi.lego2 = LegoYapBenzersiz();  else legoHepsi.lego2 = LegoYap();      }    public LegoOrtak LegoYap(int sec)  {  LegoOrtak temp;  if (sec == 0) { temp = new LegoO(); }  else if (sec == 1) { temp = new LegoI(); }  else if (sec == 2) { temp = new LegoS(); }  else if (sec == 3) { temp = new LegoZ(); }  else if (sec == 4) { temp = new LegoL(); }  else if (sec == 5) { temp = new LegoJ(); }  else { temp = new LegoT(); }  return temp;  }  //Random lego  Random rnd = new Random();  public LegoOrtak LegoYap()  {  int sec = rnd.Next(0,7);  return LegoYap(sec);  }  public LegoOrtak LegoYapBenzersiz()  {  etiket:  int sec = rnd.Next(0,7);  LegoOrtak temp = LegoYap(sec);  if (temp.Adi == legoHepsi.lego1.Adi)  {  goto etiket;//Bu yüzden renkler benzersiz olmalı  }  return LegoYap(sec);  }  }  } |
|  |

|  |
| --- |
| SESEKLEME CLASS |
| using System;  using System.Media;  namespace GoodDuck  {  class SESEKLEME  {  private bool acikMi;  public bool AcikMi  {  get { return acikMi; }  set  {  acikMi = value;  if (value)  {  Play();  }  else  {  Stop();  }  }  }  public string ad { get; set; }  SoundPlayer player;    public SESEKLEME(string str)  {  player = new SoundPlayer();  sesYolu(str);  }  public SESEKLEME()  {  player = new SoundPlayer();  }  public void sesYolu(string str)  {  player.SoundLocation = str;  }  public void EklemeAd(string str)  {  this.ad = str;  }  public bool Play()  {  //FALSE İSE OYNATILAMADI  bool temp = true;  try  {  player.Stop();  player.Play();  }  catch (Exception)  {  temp = false;  }  return temp;  }  public void Stop()  {  try  {  player.Stop();  }  catch (Exception)  {    }  }    }  } |
|  |

|  |
| --- |
| SKORLAR CLASS |
| namespace GoodDuck  {  class SKORLAR  {  static int skorDegeri { get; set; } = 0;  public static string SkorOku { get { return skorDegeri + " Puan"; } }  public static void skorSifirla()  {  skorDegeri = 0;  }  public static void skorHesaplaBiriktir()  {  skorDegeri += 4;  }  public static void skorHesaplaSil()  {  skorDegeri += 10;  }  }  } |
|  |

|  |
| --- |
| SURE CLASS |
| namespace GoodDuck  {  public class SURE  {  public int toplamZaman { get; private set; } = -1;  public int muzikSay { get; private set; }    public void Sifirla()  {  SifirlaToplamZaman();  SifirlaMuzikSay();  }  public void EkleToplamZaman()  {  toplamZaman++;  }  public void EkleMuzikSayZaman()  {  muzikSay++;  }  public void SifirlaToplamZaman()  {  toplamZaman = -1;  }  public void SifirlaMuzikSay()  {  muzikSay = -1;  }  public void MuzikSay()  {  if (legoHepsi.SesCalisiyorMu)  {  muzikSay++;  }    }  public bool CalKekeCal()  {    bool temp = false;  if (muzikSay==81)  {  temp = true;  }  else  {  temp = false;  }  return temp;  }  public int dondurSaniye { get { return (toplamZaman); } }  public string dondurSaniyeText { get { return (toplamZaman).ToString() + " sn"; } }  }  } |
|  |

|  |
| --- |
| FRMANA CLASS |
| using GoodDuck;  using System;  using System.Collections.Generic;  using System.Drawing;  using System.Windows.Forms;  namespace GoodDuck  {  public partial class frmANA : Form  {  public frmANA() { InitializeComponent(); }  #region -------------------------------------DEĞİŞKENLER------------------------------------------  #region GÖSTERİLEN LEGOLAR  List<PictureBox> legoResimler1;  List<PictureBox> legoResimler2;  #endregion  LEGOSILME legosilme;  GOSTER goster;  LEGOYAP legoyap;  HIT hit;  SESEKLEME sesMuzik;  SURE sure = new SURE();  //SIL sil;  #endregion  #region BAŞLANGIÇ -- RESİMLER YÜKLE  private void Form1\_Shown(object sender, EventArgs e)  {  int i = 199;//    legoHepsi.resimler = new List<PictureBox>();    foreach (PictureBox item in tableLayoutPanel1.Controls)  {  item.Name = i.ToString();  i--;  legoHepsi.resimler.Add(item);  }  legoHepsi.resimler.Reverse();  legoResimler1 = new List<PictureBox>(); int i2 = 15;  foreach (PictureBox item in tableLayoutPanel2.Controls)  {  item.Name = i.ToString(); i2--;  legoResimler1.Add(item);  }  legoResimler2 = new List<PictureBox>(); i2 = 15;  foreach (PictureBox item in tableLayoutPanel3.Controls)  {  item.Name = i.ToString(); i2--;  legoResimler2.Add(item);  }  legoResimler1.Reverse();  legoResimler2.Reverse();  legosilme = new LEGOSILME() { resimler = legoHepsi.resimler, legolar = legoHepsi.legolar };  hit = new HIT(legoyap);    sesMuzik = new SESEKLEME("muzik.wav");  sesMuzik.AcikMi = false;    sure = new SURE();  for (int d = 0; d < 8; d++)  {  Resimler.resimYukle(d);  }    BASLANGIC();  }    #endregion      private void BASLANGIC()  {  //TÜM RESİMLERİ TEMİZLE  for (int i = 0; i < legoHepsi.resimler.Count; i++)  {  legoHepsi.resimler[i].BackgroundImage = Resimler.varsayilan;  }  //ŞİMDİKİ SONRAKİ RESİMLER TEMİZLE  #region HANGİ TETRİSLER VAR ONLARI GÖSTERİR  for (int i = 0; i < legoResimler1.Count; i++)  {  legoResimler1[i].BackgroundImage = Resimler.varsayilan;  legoResimler2[i].BackgroundImage = Resimler.varsayilan;  }  #endregion  legoHepsi.legolar = new List<int>();  legoHepsi.OyunCalisiyorMu = true;  goster = new GOSTER();  legoyap = new LEGOYAP();  legosilme = new LEGOSILME() { resimler = legoHepsi.resimler, legolar = legoHepsi.legolar };  sure.Sifirla();  SesCal();  timerYercekimi.Enabled = true;  timerGecenSüre.Enabled = true;  SimdikiSonrakiGoster();  goster.Goster();  SKORLAR.skorSifirla();  }        private int EskiKonumHesap(int sayi)  {  int temp = sayi;  if (sayi < 4)  {  temp = sayi;  }  else if (sayi < 14)  {  temp -= 6;  }  else if (sayi < 24)  {  temp -= 12;  }  else  {  temp -= 18;  }  return temp;  }  private void SimdikiSonrakiGoster()  {  LegoOrtak lego1 = legoHepsi.lego1;  LegoOrtak lego2 = legoHepsi.lego2;  Yapi legYapi1 = lego1.Dondur[0];  Yapi legYapi2 = lego2.Dondur[0];  for (int i = 0; i < 16; i++)  {  legoResimler1[i].BackgroundImage = Resimler.varsayilan;  legoResimler2[i].BackgroundImage = Resimler.varsayilan;  }  int a1 = 0, b1 = 0, c1 = 0 ,d1 = 0;  int a2 = 0, b2 = 0, c2 = 0, d2 = 0;  //MessageBox.Show(legYapi1.a + " ");  a1 = EskiKonumHesap(legYapi1.a);  b1 = EskiKonumHesap(legYapi1.b);  c1 = EskiKonumHesap(legYapi1.c);  d1 = EskiKonumHesap(legYapi1.d);  a2 = EskiKonumHesap(legYapi2.a);  b2 = EskiKonumHesap(legYapi2.b);  c2 = EskiKonumHesap(legYapi2.c);  d2 = EskiKonumHesap(legYapi2.d);  // MessageBox.Show(a1+" "+b1+" "+c1+" "+d1);  if (lego1 != null && lego2 != null)  {    legoResimler1[a1].BackgroundImage = lego1.resim;  legoResimler1[b1].BackgroundImage = lego1.resim;  legoResimler1[c1].BackgroundImage = lego1.resim;  legoResimler1[d1].BackgroundImage = lego1.resim;  legoResimler2[a2].BackgroundImage = lego2.resim;  legoResimler2[b2].BackgroundImage = lego2.resim;  legoResimler2[c2].BackgroundImage = lego2.resim;  legoResimler2[d2].BackgroundImage = lego2.resim;  }  else  {  MessageBox.Show("Lego Üretilmemiş");  }  //MessageBox.Show(legoYapi1.a + " " + legoYapi1.b + " " + legoYapi1.c + " " + legoYapi1.d + " ");  //MessageBox.Show(legoYapi2.a + " " + legoYapi2.b + " " + legoYapi2.c + " " + legoYapi2.d + " ");  ///MessageBox.Show(lego1.Dondur[0].a + " " + lego1.Dondur[0].b + " " + lego1.Dondur[0].c + " " + lego1.Dondur[0].d + " ");  }    private void TEST\_Baslat\_Click(object sender, EventArgs e)  {  BASLANGIC();  }    #region \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*CLİCKLER\*\*\*\*\*\*\*\*\*\*\*\*\*YÖNLERRR\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*        private void sola\_Click(object sender, EventArgs e)  {  if (legoHepsi.OyunCalisiyorMu)  {  if (legoHepsi.lego1 != null)  {  if (legoHepsi.lego1.hareketeDevam)  {  goster.Temizle();  string str = hit.solla();  // listBox1.Items.Add(str + " " + legoHit.solDuvar + " " + legoHit.solNesne);  goster.Goster();  }  }  else  {  MessageBox.Show("lego1 oluşturulmamış");  }  }  }  private void saga\_Click(object sender, EventArgs e)  {  if (legoHepsi.OyunCalisiyorMu)  {  if (legoHepsi.lego1 != null)  {  if (legoHepsi.lego1.hareketeDevam)  {  goster.Temizle();  string str = hit.sagla();  //listBox2.Items.Add(str + " " + legoHit.sagDuvar + " " + legoHit.sagNesne);  goster.Goster();  }  }  else  {  MessageBox.Show("lego1 oluşturulmamış");  }  }    }  private void asagi\_Click(object sender, EventArgs e)  {  if (legoHepsi.OyunCalisiyorMu)  {  //legosilme.OYNAT();  //AŞAĞI DOĞRU HAREKET  if (legoHepsi.lego1 != null)  {  if (legoHepsi.lego1.hareketeDevam)  {  goster.Temizle();  string str = hit.asagila();  //listBox3.Items.Add(str + " " + legoHit.asagiDuvar + " " + legoHit.asagiNesne);  goster.Goster();  }    }  else  {  MessageBox.Show("lego1 oluşturulmamış");  }  //---------------------------LEGO HAREKETİNİ DURDURDUYSA()---------------------------------  if (!legoHepsi.lego1.hareketeDevam)  {    ///---------------------------YENİLDİYSENİZ---------------------------------  if (legoHepsi.lego1.EkleLegolar())  {  timerYercekimi.Enabled = false;  sesMuzik.AcikMi = false;  legoHepsi.OyunCalisiyorMu = false;  DialogResult cikis = MessageBox.Show("Yeniden Oynamak İstiyormusunuz ?", "Yeni Oyun", MessageBoxButtons.YesNo);  if (cikis == DialogResult.Yes)  {  BASLANGIC();  }  else  {  Application.Exit();  }  }  //---------------------------LEGO SONRAKİNE GEÇ---------------------------------  legoyap.Sonraki();  SimdikiSonrakiGoster();  //-----------------------------------------------------------------------------    }  legosilme.OYNAT();    }  }  private void Dondur\_Click(object sender, EventArgs e)  {  if (legoHepsi.OyunCalisiyorMu)  {  if (legoHepsi.lego1 != null)  {  if (legoHepsi.lego1.hareketeDevam)  {  goster.Temizle();  string str = hit.dondurme();  //listBox5.Items.Add(str + " yan " + legoHit.donmeDuvarYan + " yer " + legoHit.donmeDuvarYer + " nesne " + legoHit.donmeNesne);  goster.Goster();  }  else  {  //MessageBox.Show("lego false"+legoHepsi.lego1.hareketeDevam);  }  }  else  {  MessageBox.Show("lego1 oluşturulmamış");  }  }  }  #endregion    private void frmANA\_Load(object sender, EventArgs e)  {  }  private void btnYeniOyun\_Click(object sender, EventArgs e)  {  BASLANGIC();  }  private void btnSesAcikKapali()  {  legoHepsi.SesCalisiyorMu = !legoHepsi.SesCalisiyorMu;  }  private void SesCal()  {  if (legoHepsi.SesCalisiyorMu)  {  sesMuzik.AcikMi = true;  }  else  {  sesMuzik.AcikMi = false;  }  }  private void btnSes\_Click(object sender, EventArgs e)  {  btnSesAcikKapali();  if (legoHepsi.SesCalisiyorMu)  {  sesMuzik.AcikMi = true;  //(sender as Button).Text = "Ses Açık";  }  else  {  sesMuzik.AcikMi = false;  //(sender as Button).Text = "Ses Kapalı";  }      }    private void btnPause\_Click(object sender, EventArgs e)  {  legoHepsi.OyunCalisiyorMu = !legoHepsi.OyunCalisiyorMu;  if (legoHepsi.OyunCalisiyorMu)  {  timerYercekimi.Enabled = true;  timerGecenSüre.Enabled = true;  if (legoHepsi.SesCalisiyorMu)  {  sesMuzik.Play();  }    }  else  {  timerYercekimi.Enabled = false;  timerGecenSüre.Enabled = false;  sesMuzik.Stop();  }  }  #region TİMER İNTERVAL  private void timerInterval(int interval)  {  timerYercekimi.Interval = interval;  }  private void timerInterval10Kati(int interval)  {  timerInterval(interval \* 10);  }  private void timerInterval(double interval)  {  timerInterval(Convert.ToInt32(interval));  }  private void timerInterval10Kati(double interval)  {  timerInterval(Convert.ToInt32(interval) \* 10);  }  #endregion  private void timerTetrisHizi\_Tick(object sender, EventArgs e)  {  if (legoHepsi.lego1 != null)  {  asagi\_Click(null, null);  }  }  private void timerGecenSüre\_Tick(object sender, EventArgs e)  {  if (legoHepsi.OyunCalisiyorMu)  {  sure.EkleToplamZaman();  labelSure.Text = sure.dondurSaniyeText;  if (sure.CalKekeCal())  {  sesMuzik.Play();  sure.SifirlaMuzikSay();  }  else  {  sure.MuzikSay();  }  labelSkor.Text = SKORLAR.SkorOku;  }      }    private void btnTEST\_Click(object sender, EventArgs e)  {  //legosilme.OYNAT();  }  public Keys sol { get; private set; } = Keys.Left;  public Keys sag { get; private set; } = Keys.Right;  public Keys yukari { get; private set; } = Keys.Up;  public Keys asagi { get; private set; } = Keys.Down;  public Keys pause { get; private set; } = Keys.D;  public Keys sesAcKapa { get; private set; } = Keys.S;  public Keys yeniOyun { get; private set; } = Keys.A;  private void frmANA\_KeyDown(object sender, KeyEventArgs e)  {  if (e.KeyCode == sol) sola\_Click(null, null);  if (e.KeyCode == sag) saga\_Click(null, null);  if (e.KeyCode == asagi) asagi\_Click(null, null);  if (e.KeyCode == yukari) { Dondur\_Click(null, null); }  if (e.KeyCode == pause) btnPause\_Click(null, null);  if (e.KeyCode == sesAcKapa) { btnSes\_Click(null, null); }  if (e.KeyCode == yeniOyun) btnYeniOyun\_Click(null, null);  //MessageBox.Show("Form Key Basıldı");  }  private void timerSeriPort\_Tick(object sender, EventArgs e)  {  try  {    if (!serialPort1.IsOpen)  {  serialPort1.Open();  }  goster.Goster();  string strr = (serialPort1.ReadLine());  if (strr[0]=='1') btnYeniOyun\_Click(null,null);  else if (strr[1] == '1') btnSes\_Click(null, null);  else if (strr[2] == '1') btnPause\_Click(null, null);  else if (strr[3] == '1') sola\_Click(null, null);  else if (strr[4] == '1') saga\_Click(null, null);  else if (strr[5] == '1') Dondur\_Click(null, null);  else if (strr[6] == '1') asagi\_Click(null, null);  //this.Text = strr;  }  catch (Exception ex)  {  Text = ex.Message;  }  }  //listBox1.Items.Add("lego Kayit sayisi "+legoHepsi.legolar.Count+" lego resim sayisi "+legoHepsi.resimler.Count+" class1 resimler sayisi "+class1.resimler.Count);  }  } |
|  |