



► **NORMAL TESLİM: 18 TEMMUZ PERŞEMBE 23:59'DAN ÖNCE**

- **ERKEN TESLİM ÖDÜLÜ:** Erken teslim edilen her gün için 3/100 ödül verilir. (maksimum 15/100)
- **GEÇ TESLİM:** 19 Temmuz Cuma sabah saat 08:59'dan önce (-%30* PUAN CEZA)
- **GEÇ TESLİM:** 19 Temmuz Cuma sabah saat 09:59'dan önce (-%50* PUAN CEZA)
*Öğrencinin ödev notuna referansla

Ödev duyurusu 24 Haziran'da yapılmış olup, ödevin tamamlanması için yaklaşık 24-25 gün süre verilmiştir.

```
printf("Bil141 Bilgisayar Programlama dersinin son ödevine hoş geldiniz.\n");
```

```
while ( enemy.x == x && enemy.y == y )
```

```
{
```

```
    enemy.health -= fire.power;
```

```
}
```

Detayları ilerleyen sayfalarda verilecek olmakla birlikte ödev şu şekildedir: Bir **bilgisayar oyunu** hayal etmeniz ve kodlamanız istenmektedir.

- Kullanıcıyla etkileşimli, mantıklı, akıcı... bir oyun kurgulamanız ve kodlamanız istenmektedir.
- Bu oyun SYÇS6'daki **Sayı Oyunu** ve çok basit olan **Taş Kağıt Makas** haricinde bir oyun olmalıdır.
 - SYÇS6'daki Sayı Oyununu bu ödevde ön hazırlık olarak yapmanızı öneriyoruz.
 - Taş Kağıt Makas ise hem çok basit (sizi Ara Sınav II'e hazırlama amacımıza uymuyor) hem de derste birlikte kodlama ihtimalimiz var.

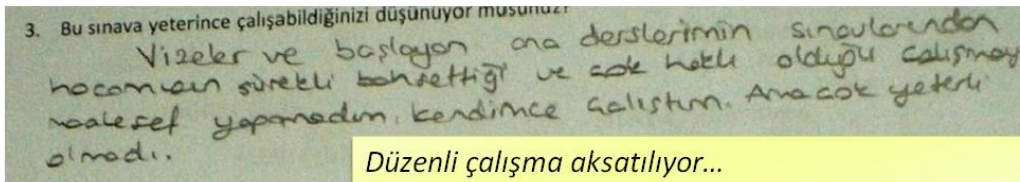
Alternatif olarak hoşunuza giden (oyun olmak zorunda olmayan) bir **proje** de yapabilirsiniz. Bunun için öncelikle aklınızdaki projeyi bize danışıp onay alınız.

Bu ödevin amacı Ara Sınav II'ye hazırlanmanıza yardımcı olmaktır.

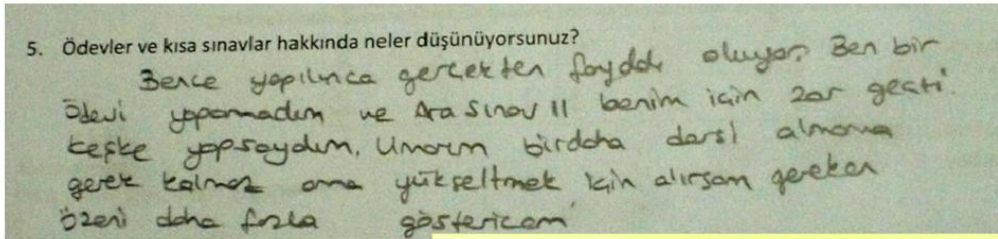
- 2018 Güz Döneminde dersi alan 285 kişiden sadece 52'si Ödev3'ü (%20) yaptı.
- 2018 Güz Dönemi Ara Sınav II ortalaması 33/100
- 2018 Güz Dönemi Ödev3'ü yapanların ortalaması, yapmayanların yaklaşık 2 katı (1,93) çıktı.
- Ara Sınav II'de 80 üstünde alanların tamamı Ödev3'ü yapmıştı.
- 2018 Güz Döneminden bazı ödev geri bildirimleri:
 - *Potansiyeli çok büyük bir ödev sınavdan sonraya bıraktığım için pişmanım. Daha gelişmiş yapabilirdim.*
 - *Ödevle keşke daha çok uğraşabilseydim... o zaman hem kendi adıma daha da faydalı olurdu...*
 - *Ödev gerçekten c'yi kavramama yardımcı oldu. Çok iyi bir pratik oldu benim için. Keşke böyle ödevler daha çok olsa...(puan olmasa da olur:)*
 - *Kendi yaptığım bir oyunu oynamak çok güzel, kardeşime de oynattım çok beğendi. Ödev, dersi pekiştirmek, özellikle fonksiyonları kullanmak açısından çok etkili oldu.*
 - *Eğlenceliydi... Artık bilgisayar başında beklemem gerekirse bu oyunu oynayarak eğlenceli zaman geçirebilirim*
 - *Ödev gerçek manada öğretici oldu.Güzel bir ödevdi.*
 - *Günlük ortalama 2 saat harcadım.Zaman oldukça ortaya çok güzel şeyler çıkabiliyor.Ne yazık ki kısıtlı zamanlarda(sınavlar gibi) çok da verimli olamıyorum. Bu konuda kendimi hızlandırmam gerektiğini anladım.*
 - *Özellikle ekrana çizme konusunda çok gelişmemi sağladı.*

► Güz 2018 döneminden AA almayı seçmemiş bir öğrenci...

► Sınav notları 73/35/70 – Harf notu CB



Düzenli çalışma aksatılıyor...



Ödev3 ile ara sınav II'e hazırlanma imkanı kullanılmıyor.

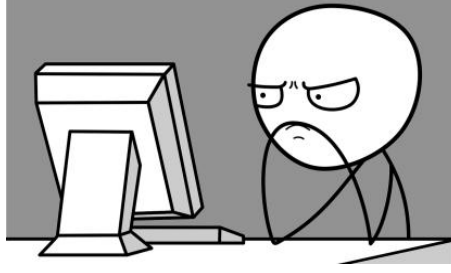


Şekil 1 - Bu ödevde hayal etmenizi ve ilk başta **"benim bunu yapmam imkansız"** seviyesinde gözükten bir fikri **"zoru yaparız, imkansız biraz zaman alır"** anlayışıyla yapmanızı diliyoruz.

Sınırlarınızı zorlamazsanız gelişebilir misiniz?

- Kodlamadaki anahtar yaklaşım olan **BEBEK ADIMLARIYLA** ilerlemeye bu ödevde de büyük özen gösteriniz.
 - Kodunuza eklemeyi düşündüğünüz bir özelliği önce başka bir sekmede tek başına tasarlayın, test edin ve ondan sonra kodunuza ekleyiniz. Hepsini bir arada yapmaya kalkmanız vakit kaybına yol açabilir.
 - Kendinize **ARA HEDEFLER** belirleyin, o hedeflere erişin, sonra bir sonraki hedefe yönelin.
- Düşüncelerinizi **ÖNCE SÖZDE KODA** dökmek, sonra C koduna geçirmek iyi bir stratejidir.
 - Şöyle düşünün: *Benim bu ödevde ayırabileceğim vakit X saat. Bu X saati aşarsam diğer derslerime ayıracağım vakitten almış olacağım. Hem bu X saati aşmamalı hem de bu X saatte beni Ara Sınav II'e hazırlayacak kaliteli bir iş ortaya çıkarmalıyım.*
 - Bunun için **vakti** verimli **KULLANMALISINIZ**. Sözde kod kullanmadan da bu kodu yazarsınız, ancak mühendislikte en önemli parametrelerden biri olan ZAMANı verimli kullanma gayeniz varsa **SÖZDE KOD KULLANINIZ**.
- Kodunuz çalışmadığında **DÜŞÜNÜN**. Bunu yapabilmek adına, ellerinizi klavyeden çekin, arkanıza yaslanın ve kendinizi **DÜŞÜNMEYE ZORLAYIN!**

Neden benim istediğimden farklı şekilde çalışıyor olabilir?



Şekil 2 - Kodunuz çalışmadığında ellerinizi klavyeden çekin ve düşünün.

- Sık yapılan ve yanlış bir yaklaşım olan **KAŞ YAPARKEN GÖZ ÇIKARMAK**'tan özenle kaçınınız! Kodunuz çalışmadığında bilinçsizce kodun sağını solunu bozmak, anlamsız denemeler yapmak, sizi **ara sınav II'ye iyi şekilde HAZIRLAMAYABİLİR** ve kısıtlı olan çalışma **vaktinizden yer**.
 - Kodunuz "nedensizce" çalışmıyor olduğunda bize danışmaya çekinmeyin.
 - Ofis saatlerini es geçmeyin.



Şekil 3 - Kendinizi **DENEME YANILMA GİRDAB**ından uzak tutun.

- **Deneme yanılma girdabına** düşmeye meyilli bir öğrencinin kodu çalışmadığında ikisi de faydasız olan şu durumlar başına gelebilir.
 - Kodun sağını solunu bozar ve kod çalışmaz (genelde olan).
 - Buzdolabına gider, bir şeyler alırken aklına bir şey daha gelir.
 - Onu da dener ve kod çalışmaz.
 - TV'ye göz atarken aklına bir şey daha gelir.
 - Onu da dener ve kod çalışmaz.
 - Başka derse çalışırken... (kaybolan saatler)



Şekil 4 - Esrarengiz şekilde çalışan kodlar nankördür.

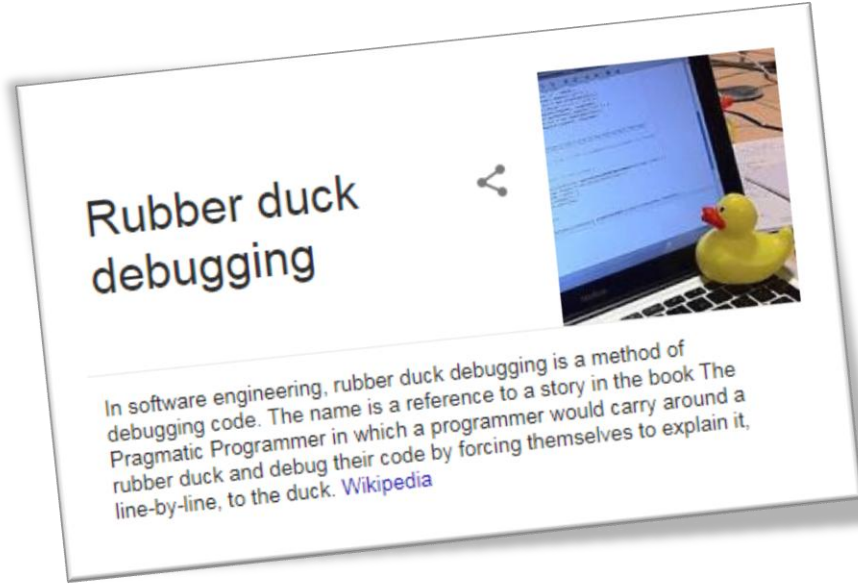
- Kodun sağını solunu bozarken bir aşamada kod çalışır...
 - a) Ancak başka yerleri bozmuştur ve geri düzeltemez.
 - b) Esrarengiz şekilde çalışır hale geldiği için, "yine bozulursa" korkusuyla artık kodu pek de geliştirmek istemez.
 - c) Kod çalışır gibi gelmiştir ama aslında değerlendirme aşamasında asistanın bilgisayarında çalışmayacaktır çünkü kodda dizilerin dışına kayıt yapılması gibi önemli hatalar olmuştur.

TÜM DURUMLARDA ORTAK OLAN BİR ŞEY VAR Kİ:

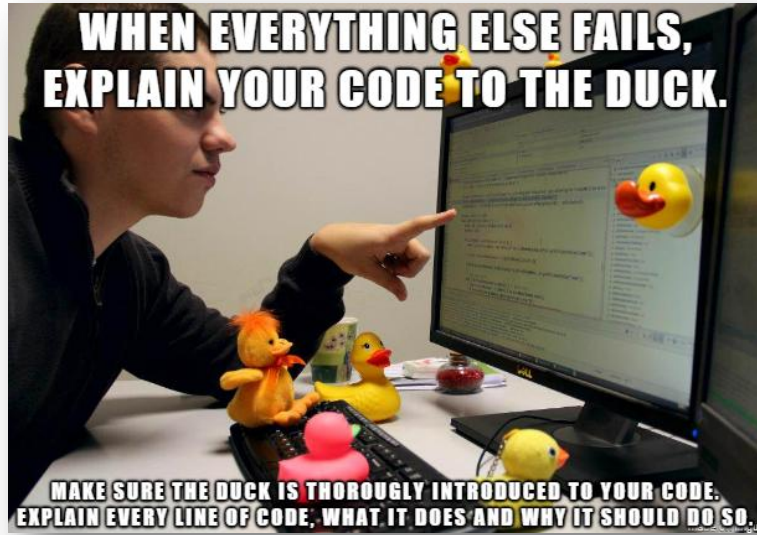
BİLİNÇSİZCE DENEME YANIMA SÜRECİ

SİZİ ARA SINAV II'YE YETERLİ ŞEKİLDE HAZIRLAMAZ.

- Kodunuzu geliştirirken bir yerde tıkanıldığınızda ya da koddaki hatayı bulamadığınızda *Homosapiens(insan)* olmayan birine kodunuzu anlatın. (*Homosapiens*'lere kod anlatımının sakıncaları için bkz. Piazza'daki "*Odevlerdeki_Etik_Ilkelerimiz.pdf*")

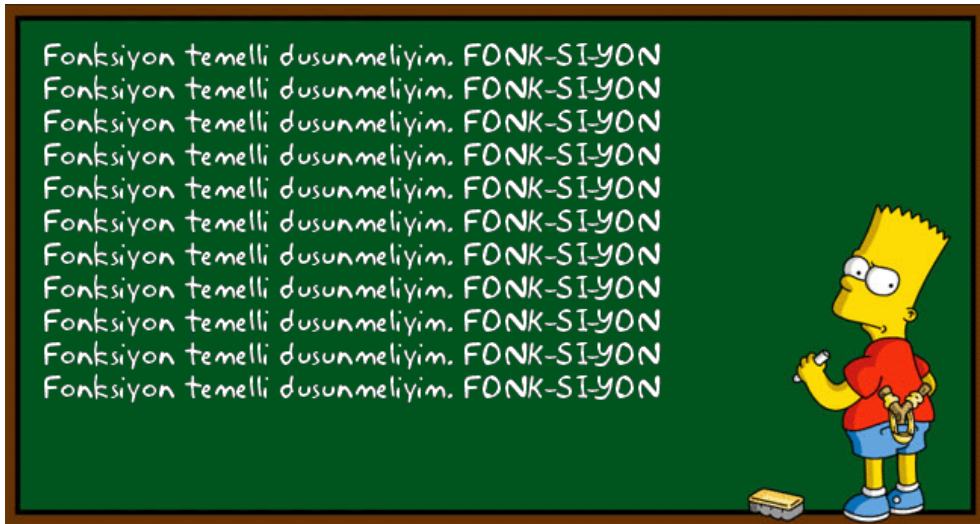


Şekil 5 - Programlamadaki etkin yöntemlerinden biri lastikten bir ördeğe kodunuzu satır satır anlatmaktır. *"İşte burada da fonksiyondan gelen değeri... bir dakika... ben niye onu kaydetmedim."* şeklinde bir aydınlanmaya yol açabilir.



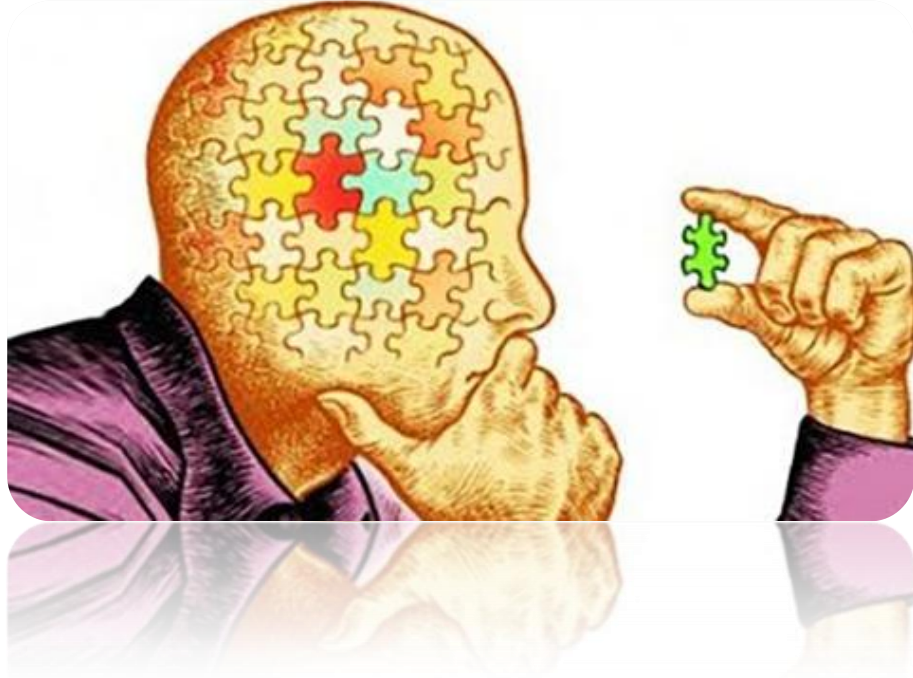
Şekil 6 - Ara sınav II'ye lastik ördek getirebilirsiniz ama dalgınlıkla sıkıp ses çıkartarak ya da ördekle seslice konuşarak diğer öğrencileri rahatsız etmeyiniz.

- **FONKSİYON** temelli düşünün. Daha da açıkçası, FONKSİYON TEMELLİ DÜŞÜNMEYE **KENDİNİZİ ZORLAYIN**.
 - Hayal kütüphaneniz olsun ve bu kütüphanenin içinde hayal ettiğiniz gibi bir fonksiyonun olduğunu hayal edin.
 - Kodunuzun başında *#include <hayal.h>* varmış gibi düşünün...
 - "Eğer elimde *şöyle bir fonksiyon* olsaydı, ben bu kodu aslında çok rahat kodlardım." şeklinde düşünmeye kendinizi zorlayın ve "*şöyle bir fonksiyon*"unuzun ne olduğuna karar verin. Bu fonksiyonu **SÖZDE KOD** ile kağıt üzerinde tasarlayın, ayrı bir sekmede kodlayın, iyice test edin ve sonra güle güle kullanın.
 - Anı: Kod doğrulayıcıyı kodlarken windows cmd'de "more" isimli hazır bir fonksiyonun olduğunu fark ettim ve bu fonksiyon kodumu oldukça rahatlatmıştı. "*İyi ki bu fonksiyon varmış*" şeklinde düşünürken aslında bu fonksiyonu rahatlıkla kendim de kodlayabileceğimi fark ettim. Fonksiyon temelli düşünmemiş ve main'e fazla yüklenmiştim.
 - main'e fazla yüklenmek kodun geliştirilebilir olma özelliğinden çalar ve bir yerden sonra (gelecekte işyerinizde yazdığınız kodda) kodunuz için "*lütfen koduma dokunmayınız, çalışana kadar içim dışıma çıktı, bu kodu olduğu gibi kabulleniniz ve geliştirmeye kalkmayınız. Bir satırına bile dokunmayınız.*" duygusuna kapılırsınız. Kendinize bir kötülük yapmak istiyorsanız, "*esrarengiz şekilde çalışan kod*" yazınız. Böylece yazacağınız her kodda sıfırdan başlamak zorunda kalırsınız, taş üstüne taş koyamazsınız ve piyasada ağır kalırsınız.

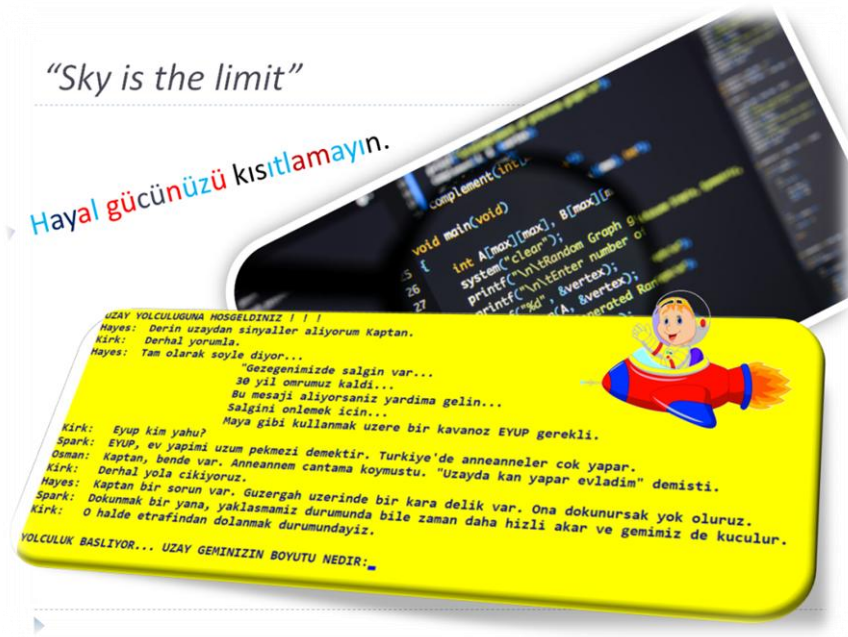


Şekil 7 - Hedefiniz kendinizi iyi bir programcı olarak yetiştirmekse kendinizi fonksiyon temelli düşünmeye zorlayın.

- Bilgisayarlar yarı iletkenler üzerine inşa edilmiş *transistör* isimli devre elemanlarının birleşerek mantıksal kapılar oluşturmaları, bu mantıksal kapıların birleşerek tümleşik devreler (VLSI ile toplayıcılar, tümleyiciler ...) , onların da birleşerek ... saniyede birkaç milyar işlem yapan bir alet oluşturmalarıyla ortaya çıkmıştır. Birden *bilgisayar* yapacağım diye ortaya çıkan *yerinde sayar*. Ancak **parçalara ayırarak ilerlenilirse** bilgisayar gibi harika bir aleti ortaya çıkarmak, tasarımını anlamak ve geliştirmek mümkün olur.
- Analitik düşünmenin (*bir problemi sade ve çözülebilir parçalara ayırabilme, parçaları çözüp, sonuçları birleştirebilme*) düşünce yapısının kodlamadaki karşılığı **fonksiyonlardır**. Tüm işleri main'e yaptırmaya çalışırsanız bir süre sonra tıkanırsınız. Detaylarla uğraşmaktan ufku göremezsiniz. Fonksiyon temelli düşünmek **ilk başta zor gelebilir**, ama zamanla bu şekilde sistemli kodlama ile gelişmiş kodlar ortaya çıkarabilirsiniz. Hem kodlarınızın gelişmişliğine hem de harcadığınız eforun göreceli olarak azlığına baktığınızda kendinize hayran kalırsınız ve *"aslında o kadar da zor değilmiş"* dersiniz.
 - Nasıl ki aynı düşünce yapısına sahip ve iyi bildikleri dilde konuşan iki insan birbirleriyle rahatlıkla anlaşabilirler; siz de bilgisayar gibi düşünebilerseniz, derdinizi onun dilinde KISA, ÖZ ve AKICI olarak anlatarak onunla rahatlıkla anlaşabilirsiniz.



Şekil 8 - Analitik düşünmeyi bir alışkanlık haline getiriniz.

"KENDİ KODLADIĞI BİLGİSAYAR OYUNUNU OYNAMA HİSSİ... VAZGEÇİLMEZ"

Şekil 9 - Hayal gücünüzü zorlayın

Oyununuz algoritmik seviyesine göre şu şekillerde değerlendirilir:

- **ÇOK BASİT (40 PUAN):** Çok sade ödevler. Böyle bir ödevi reddetmeyecek, değerlendirmeye katacağız ama sizi sınava yeterli şekilde hazırlamayabilir! Örneğin şayet taş-kağıt-makas kabul ediliyor olsaydı "Çok Basit" olurdu.
- **BASİT(70 PUAN) :** Derste öğrendiklerimizi içeren ama çok iyi kodlanmamış, parametrik düşünceyi içermeyen, gelişime açık olmayan bir ödev. Örneğin şayet Sayı Oyunu kabul ediliyor olsaydı SYÇS6'daki "Temel Versiyon" bol bol if else ile yazılabileceği için bu seviyede değerlendirilirdi.
- **NORMAL(100 PUAN):** Oyun kurgusunun gerektirdiği C yapılarını (döngüleri, dizileri, fonksiyonları...) içeren ve algoritmik olarak yeterli karmaşıklıkta bir kod. Örneğin şayet Sayı Oyunu kabul ediliyor olsaydı SYÇS6'daki "Gelişmiş Versiyon" yoğun bir dizi, döngü... kullanımı ve iyi bir algoritmik kurgu gerektirdiği için bu seviyede değerlendirilirdi.
- **GELİŞMİŞ(100 ÜSTÜ PUAN - MAKS. 200):** Normal için koyduğumuz seviyenin ötesine geçmeyi başarmış bir ödev. Ancak bunun için vasat bir oyunun ötesini hedefleyiniz. Ara Sınav II'ye sizi hazırlayabilmesi için sınırlarınızı zorlayacak bir oyun tasarlayınız.

Ödevinizde **kullanıcının ok girişlerini** almanız şart değildir. Ama kurgunuz gerektiriyorsa, "**kullanıcının ok girişi nasıl alınır?**" sorusu gündeme gelir.

- Öncelikle scanf'ı unutunuz.
 1. Kimse "sağ ok-enter-sol ok-enter..." şeklinde oyun oynamak istemez.
 2. Zaten scanf ok tuşunu alamaz ama "a","s","d" tuşlarıyla oynatacak olsanız bile ekranda "asassaaaaasssssd" şeklinde girişlerin gözükmesi de ayrı bir dert olur.
- getch() girişin ardından enter'a basma gerektirmez ve basılan tuş ekranda gözükmez.
 1. getch_denemem.c 'i oluşturunuz.
 2. conio.h'ı ekleyiniz.
 3. printf("%d", getch()); ile getch'i inceleyiniz.

```

1  #include <stdio.h>
2  #include <conio.h>
3  int main()
4  {
5      while(1==1)
6      {
7          printf( "%d\n", getch() );
8      }
9      return 0;
10 }
11

```

Execution output: 224, 77

Şekil 10 - Önce ayrı sekmede getch 'i ok tuşlarıyla kullanmayı öğreniniz. Sağ ok tuşuna basıldığında ekrana ne yazdırıyor?

- SUNU02'nin sonundaki getch slaytlarını inceleyiniz.

getch() ile ok girişi algılamak

Ok yönleri	Soldaki ok tuşları	Sayı kısmı üzerindeki
Üst	224-72	0-72
Alt	224-80	0-80
Sol	224-75	0-75
Sağ	224-77	0-77

- Ok tuşları bir değil iki tane int değeri ile temsil edilir.
- Bunun için iki tane peş peşe getch() gereklidir.
 - Birinci değer ok tuşlarının klavyeden girildiği yeri gösterir.
 - İkinci değer ok tuşunun yönünü gösterir.

► F1(0-59), Del(224-83)... gibi tuşlar da iki int ile temsil edilir.

Şekil 11 - Ok girişleri ve ürettikleri değerler

- Oyunu oynarken kullanıcıdan gelecek girişler: 224-72-224-80-224-75 şeklinde olacağı için 224'leri boş bir getch() ile almayı düşünebilirsiniz. Ancak kullanıcı bir harfe basarsa senkronizasyon bozulacağı için şu tarz bir algoritma önerilir:
 1. getch ile değer al. (x değişkenine kaydet)
 2. Alınan değer(x) 224 değilse, Git 1.
 3. getch ile değer al. (y değişkenine kaydet)
 4. y'e bakarak girişi yapılan ok yönüne karar ver.

Ödevinizde **imleci ekranda bir yere götürmeniz** şart değildir. Ama ekrana şekiller çizdirirken bu yetenek büyük avantaj sağlar. Örneğin, ekrana bir tank çizdirmek istiyorsunuz ve bu tank zamanla ekranda hareket edecek.

- Hareketli bir parçayı ekranda göstermek için ilk akla gelen, ekranı temizlemek (`system("cls")`) ve sonra tüm ekranı tekrardan basmak olsa da bu kırık bir görüntü oluşturacak ve gözü rahatsız edecektir.
- Bunun yerine şu tarz bir algoritma önerilir:
 1. Ekrandan silinmesini istediğin yere imleci götür
 2. `printf` ile ekrana boşluk bas
 3. Silinecek şeyler bitene kadar Git 1.
 4. Ekrana eklenmesini istediğin yere imleci götür
 5. `printf` ile ekrana ilgili karakteri(örn. "*") bas
 6. Eklenecek şeyler bitene kadar Git 4.

"İmleci ekranda istediğim yere nasıl götürürüm?" sorusu gündeme gelebilir.

- SUNU04'ten bilgi alınız.
- Ayrı bir sekmede basit denemelerle `imlec(x,y)` kodunu iyice test ediniz.

Ekranda gezindiren fonksiyon: `imlec(x,y)`

- ▶ Doğrudan koordinatları kullanarak ekranda gezinebilirsiniz.
- ▶ `imlec_xy(10,5); getch(); printf("Merhaba");`

**İMLECİ EKRANDAKİ VERİLEN KOMUTA GÖNDERİR.
(windows.h kütüphanesini gerektirir.)**

```
void imlec_xy(int x, int y) {
    COORD coord;
    coord.X = x;
    coord.Y = y;
    SetConsoleCursorPosition(GetStdHandle(STD_OUTPUT_HANDLE), coord);
}
```

Şekil 12 - SUNU04'ten bir slayt

"İmlecin ekranda nerede kaldığını nasıl öğrenirim? Ekranda belli bir konumda hangi karakter olduğunu nasıl öğrenebilirim?" gibi sorular gündeme gelebilir.

- Kolay değil. Bunun için bambaşka kodlarla bu işe özel pencereler açmak gerekebilir.
- *İpucu: Belki de imlecin nerede kaldığını veren bir fonksiyona ihtiyacınız yoktur. Belki de imlecin yerini zaten biliyorsunuzdur. Sonuçta imleci oraya kodunuzla siz götürmüştünüz... Benzer şekilde hangi konumda hangi karakterin olduğunu da biliyor olabilirsiniz çünkü o karakteri o konuma `printf` ile siz bastırmıştınız.*

Ödevinizde **kodunuz çalışırken bir yandan da klavyeden giriş olup olmadığını takip etmeniz** şart değildir. Ama kullanıcı giriş yapmadığı sürece ekranda değişikliklerin olması ve kullanıcının klavyeden bunlara müdahalesi şeklinde bir oyun kurgulanmış olmanız **"hem kod çalışıp hem de klavyeden giriş alabilir mi?"** sorusunu gündeme getirebilir. Örneğin, mermiler sürekli ilerlerken, klavye ile tankın kontrol edilmesi gerekebilir.

- Bunun için ilk akla gelen eş zamanlı çalışan kod parçaları oluşturmak (bkz. *SUNU05'teki thread bilgilendirmesi*) olabilir. Ancak bu ileri düzey bir konudur ve daha sade olan **kbhit** fonksiyonu ile de çözüm üretilebilir. **kbhit** fonksiyonunu daha iyi tanımak için SUNU04'teki ilgili slayttaki kodla incelemelerinize başlayabilirsiniz.

"keyboard hit" kbhit fonksiyonu

- Klavyeden bir değer girilmediği sürece kod belli bir işlemi yapsın, değer girildiğinde başka bir işleme geçsin isteniyorsa conio kütüphanesinden kbhit() kullanılabilir.
- Giriş yoksa 0, varsa sıfır harici int değer döner.

```
#include <stdio.h>
#include <conio.h>

int main() {
    while (!kbhit())
        printf("Giris yapilmadi.\n");

    printf("%c girisi yapildi.", (char)getch());

    return 0;
}
```

Bu yöntemle birçok ilginç oyun yapılabilir!



```
Giris yapilmadi.
Giris yapilmadi.
Giris yapilmadi.
Giris yapilmadi.
Giris yapilmadi.
Giris yapilmadi.
Giris yapilmadi.
Giris yapilmadi.
Giris yapilmadi.
Giris yapilmadi.
Giris yapilmadi.
Giris yapilmadi.
Giris yapilmadi.
Giris yapilmadi.
Giris yapilmadi.
Giris yapilmadi.
Giris yapilmadi.
a girisi yapildi.
Process exited after 0.409s
Press any key to continue
```

Şekil 13 - SUNU04'ten kbhit ile ilgili bir slayt

- kbhit ile şu tarz bir algoritma düşünülebilir:
 - Oyun oynanırken çalışacak döngüyü başlat. (oyun kareleri - oyun terminolojinde sıkça geçen *fps(frame per second)* kavramı aslında ekrana basılacak bu kareler ile ilgilidir)
 - Klavyeye basılmış ise (kbhit 1 dönüyorsa), gereken güncellemeleri (örn. tankın konum parametrelerinin güncellenmesi) yap.
 - Oyunun genel güncellemesini yap (örn. mermilerin konum parametrelerinin güncellenmesi)
 - Oyun bileşenlerinin (örn. tank ve mermilerin) imlec_xy ile ekrana bastırılması
 - Git 2. (bir sonraki kareye/"frame"e geçiş)

kbhit ile eş zamanlı olmasa da eş zamanlı hissi uyandıracak şekilde oyununuz tasarlanabilir. İşlemler çok hızlı çalıştırılacağı için ve ekran güncellenme aralığında durum gerçekleşeceği için kullanıcının bakış açısından işlemler aynı anda yapılmış gibi gözükecektir.

- Döngünüz içinde **Sleep(500)** varsa, klavyeye basılsa bile önce bu Sleep'in bitmesi beklenenecektir. Bu durum oyununuza **"klavyeye basıldığında hemen tepki vermiyor"** hissi uyandırıyor çözüm düşününüz. Bunun çözümü vardır. `if(dusunmek== 1 ) var = 1;`
 - İpucu: Sleep(500) aslında yaklaşık 500 tane Sleep(1)'dir.

BAZI FİKİRLER:

- Tetris
- Tenis
- Tenis ile blokları patlatmaca
- Araba yarışı
- Ping pong
- Pacman
- Labirent çözümü
- Büyük boyutlu xox ya da sos oyunu
- Amiral battı
- Senaryolu oyun
- Kelime oyunları
- Atari tank oyunu
- Super mario
- Yılan
- 2 kişilik satranç(bilgisayara karşı oynamalı zor gelebilir)...



Bu oyunlar sadece fikir amaçlıdır. Bunlara bağlı kalmak zorunda değilsiniz. Dilerseniz fikirlerinizi bize danışabilirsiniz.

Hatırlatma: Tasarım şeklinize ve eklediğiniz özelliklere göre, “çok basit”, “basit”, normal” ya da “gelişmiş” aralığında değerlendireceğiz.

Hatırlatma: Ödevleriniz için ek sunuma çağırılırsanız ve ödevin sunumunda başarısız olursanız 5. Etik İlke uygulanır.

Etik İlke 5: “Ödevimin ek sunumunun istenebileceğini ve sunum esnasında kodumdaki bazı parçalara hakim olmadığımı anlaşırsa etik ilkeleri ihlal etmiş sayılabileceğimi biliyorum.”

Ödev ek sunumu

•Kodunuzdaki her parçaya **hakim olun**. Benzerlik testi ve Ara Sınav I sonuçları göz önünde bulundurularak bazı öğrenciler ödev sunumlarına çağırılır. Sunuma çağırılırsanız, size sorulacak sorulara rahatlıkla cevap verebilir olmalısınız.



•**Etik ilkelerin** bilincinde olarak ödevlerden **bolca faydalanın**. Deneme yanılma yöntemi(genelde ödevi son ana bırakan öğrencilerde görülen kötü bir alışkanlık) ile değil de **anlayarak kodunuzu** geliştirin ve böylece sınavlara iyi bir şekilde hazırlanın.

Hatırlatma: Ödevlerinde internet kullanımı hakkında Piazza'daki "Odevlerdeki_Etik_Ilkelerimiz.pdf" dosyasını detaylıca inceleyebilirsiniz.

Ödevlerde internet kullanımı hk. (11. ilkenin detaylı şekli)

1. Ödevde dair doğrudan araştırma(örn. “X yapan kod nasıl yazılır?”) **yapmanıza izin verilmez**.
 - Bu tarz sitelerden kod aldığınız tespit edilirse (sizin gibi başka bir öğrenci de alıp gönderdiğinde iki kod benzerlik testine takılabilir ya da ek sunum aşamasında ortaya çıkabilir vs.) etik ilke ihlali durumu yaşanabilir.
2. İnternette ancak **genel konuları** (“C’de döngüler” gibi) araştırmanıza izin verilir.
 - Bu araştırmanızda kullandığınız kaynakları da ödev formunda **açıkça belirtmeniz beklenir**. “youtube” vs. gibi ifadelerle geçiştirmeniz istenmez.
3. Bu kısıtlamaya istisna oluşturan bir durumunuz varsa, **araştırma yapmadan önce** dersin hocasına danışınız.

- Ödevlerdeki Etik İlkelerimize uymamanızı engelleyen ya da aklınıza takılan bir durum olduysa bana danışınız. Gerekliyse (ödevin gecikmeli gönderim süresinin sonuna kadar) bana e-posta atarak ödevinizi geri çekebilirsiniz.

SDL2 HAKKINDA

- Komut ekranının ötesine geçerek daha güzel (gerçek bir bilgisayar oyunundaki gibi) ekran çıktılı bir oyun yapabilmek için
- '*'larla ifade edilen görseller yerine Scratch'deki gibi kuklalar, renkler, fare girişi algılama vs. yapabilmek için ödevde erken başlamanızı ve SDL2 ile uğraşmanızı öneririz.



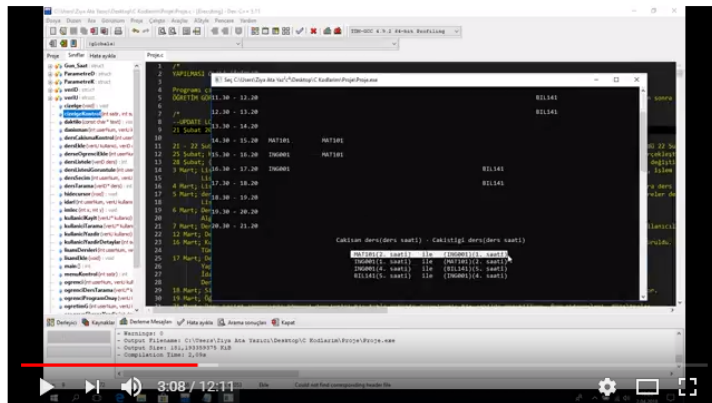
İnternette *graphics* ve *allegro* gibi kütüphaneler olsa da bu kütüphaneler güncelliğini yitirmiş ve yeni işletim sistemlerinde zor çalışır hale gelmiştir. Bu açıdan görsellik kütüphanesi olarak nispeten güncel olan SDL2'i öneriyoruz.



- Kurulum için: <http://lazyfoo.net/tutorials/SDL/> sitesindeki Code Blocks için olan yönlendirmeleri kullanınız.
 - Ufak bir detay: kodlarda `int main()` değil de `int main(int argc, char* argv[])` kullanınız.
- Öğrenmek için: CS50 dersinin <https://www.youtube.com/watch?v=yFLa3ln16w0> adresindeki seminerini ve örnek kodlarını inceleyiniz.

ALTERNATİF PROJE HAKKINDA

- <https://youtu.be/0UPG-26pIsE> adresindeki gibi **bir ders kayıt sistemi** düşünebilirsiniz.



TOBB ETU - BIL141 (Ders Kayıt Sistemi Projesi)

203 views

13 1 SHARE ...

- Eğer iyi bir nedeniniz varsa, bizi ikna edebilir ve **C harici bir dil** (C#, Python, Arduino) kullanarak da ödevinizi hazırlayabilirsiniz.
- Bir cep telefonu yazılımı, etkileşimli bir internet sitesi... gibi fikirlerinize açığız.

TAKIM ÇALIŞMASI HAKKINDA

- **Takım çalışması** yapmak isteyen öğrenciler oluşturacakları takım ile birlikte (*maksimum 3 kişilik takım*) ödevi yapabilirler.
- Bu öğrenciler önceden odamıza uğrayarak fikirleri ve takım elemanları arasında iş dağılımı hakkında bizi bilgilendirmeli ve bizden **izin almalıdır**.
- Takım çalışmasında gönderilen ödevlerde her öğrenci, katkısına göre notlandırılır. Öğrencilerin katkılarını detaylıca belirlemek üzere bu tür **ödevlerin sunumu zorunludur**.
- Ortaya çıkan çalışma öğrenci sunumlarının ardından notlandırılır ve alınan not **ödevde sunulan katkı** göz önünde bulundurularak takım elemanlarına **bölüştürülür**.
- Takım çalışması sonucu ortaya çıkan ödevlerden tüm elemanların **GELİŞMİŞ seviye notu alabilmesi** için , ortaya çıkan çalışmanın bireysel çalışmalardan **çok daha üstün** olması beklenir. Bu tür, **olağanüstü bir çalışma ortaya çıkarabilecek**, öğrencilerin bu imkandan yararlanması önerilir.
- **Tek başına çok ilerleme kaydedemeyeceğini düşünen, bireysel yapma durumunda ödevi hiç yap(a)mayacak öğrenciler** takım oluşturabilir. Ortaya çıkan çalışma vasat bir çalışma bile olsa ve ödev notu karşılığı az bile olsa takım elemanları ödev sayesinde birbirlerine destek olarak **konu eksiklerini kapatmış** olabilirler.



Her *printf* bilen C dilini biliyorum diyebildiği bir ortamda, zamanında yazmış olduğum bu kapsamlı kod size benim hakkımda bilgi verecektir.

Bu sorudan beklentimiz bir mühendiste olması beklenen (internette bkz. MÜDEK program çıktıları) "Karmaşık bir sistemi, süreci, cihazı veya ürünü gerçekçi kısıtlar ve koşullar altında, belirli gereksinimleri karşılayacak şekilde tasarlama becerisi; bu amaçla modern tasarım yöntemlerini uygulama becerisi."ne ve "Mühendislik uygulamalarında karşılaşılan karmaşık problemlerin analizi ve çözümü için gerekli olan modern teknik ve araçları geliştirme, seçme ve kullanma becerisi; bilişim teknolojilerini etkin bir şekilde kullanma becerisi"ne uygun şekilde programlamanın birçok temel kavramını kullanarak bir tasarım ortaya çıkarmanızdır. Bu tasarım sürecinin Ara Sınav II'e hazırlanmanızda ve genel olarak ise mühendisliğe hazırlanmanızda yardımcı olmasını diliyoruz. Ayrıca, hem kendinize hem de ileride kariyerinizde muhatabınıza referans olabilecek nitelikte güzel bir çalışma ortaya çıkarmanızı ve "portfolio"nuza ekleyebilmenizi diliyoruz.

- ▶ Kodunuzda goto ya da küresel değişken **kullanmayınız**. Böyle bir kullanım çok fazla puan kaybettirebilir.
- ▶ Kodunuzun **anlaşılır** olması (örn. değişken isimlerinin anlaşılır şekilde konulması), yorumlarla açıklanmış olması, girintilerinin düzgün yapılmış olması kodunuzun kalitesini artırır.
- ▶ Bir programcı olarak gelecekte ekip arkadaşlarınızın da kodunuzu anlayabilmesi için kodunuza **yorumlar** koyma alışkanlığını kazanınız. Bu size önemli gelmeyebilir ama kodunuz belirli bir büyüklüğü aştığında çok önemli olmaktadır. Aynı kod üzerinde birkaç sene sonra da çalışmanız gerekebilecek ve kodu zamanla unutacağınız için yeniden hatırlamada yorum kısımları yardımcınız olacaktır. Bu nedenle lütfen kodunuzu bol yorumlarla **ANLAŞILIR** tutun.
- ▶ Bir mühendis adayı olarak **tam olarak çalışan bir ödev** göndermeniz beklenmektedir. Gerçek hayatta kimse bozuk çalışan bir koda para vermek istemez. Ayrıca gerçek hayatta programı istenenin dışında (çalışmayan, eksik vb.) bir biçimde sunmak akademik ya da iş hayatınızda güvenilirliğinizi zedeleyebilir. Bu yüzden kodunuzu iyice test etmeli, değişik değerler ile deneyerek hatalarını ayıklamalı ve çıktıları dikkatlice inceleyerek koddaki beklenmedik herhangi bir hatanın önüne geçmelisiniz.
- ▶ Soruda satır ya da karakter limiti yoktur. Kod gönderim formuna kodunuz sığmazsa formun içine ayrı bir dosya olarak yükleyebiliyorsunuz.



**KODUNUZU DETAYLANDIRILMIŞ UYARI AYARLARI
AÇIK ŞEKİLDE
KONTROL ETTİKTEN SONRA
AŞAĞIDAKİ ADRESTE YER ALAN FORM ARACILIĞIYLA
GÖNDERİNİZ.**

<https://tinyurl.com/y4tpst5h>

- Piazza'daki ödev metni sıklıkla olmamakla birlikte kod gönderim adresinin düzeltilmesi, soru metninin daha açık hale getirilmesi gibi nedenlerle güncelleştirilebilmektedir. Bu durumlar duyuru ile sizinle paylaşılır. Güncel ödev metni **Piazza->Resources** altından erişilen metindir.
- Soruda **satır ya da karakter limiti yoktur**. Kod gönderim formuna kodunuz sığmazsa formun içine ayrı bir dosya olarak yükleyebiliyorsunuz. Ancak kodunuz gereksiz yere uzunsa muhtemelen fonksiyonları ve döngüleri etkin olarak kullanmamışsınızdır.
- Kodunuza dair özellikler için gerekiyorsa toplamda **maksimum 10 MB büyüklüğünde, en fazla 10 dosyayı** da kodunuza ek olarak yükleyebilirsiniz. Gönderim formunda bu dosyaları yükleyebileceğiniz bir kısım mevcut olacaktır. Dosya adedi ve boyutu konusundaki **limiti .zip 'leme ile artırabilirsiniz**.

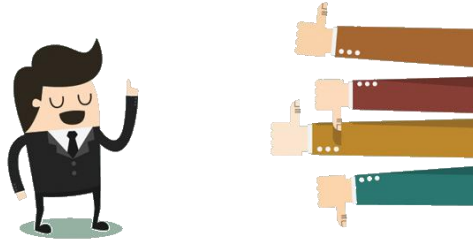
Ödevlere erken başlamanın avantajları:

1. Derste, ilgili konularda **daha etkin** olursunuz. (Eğer ödev konusu henüz derste işlenmemişse de diğer kaynaklardan öğrenmeye çalışınız. Dersi beklemeyiniz.)
2. Takıldığınız yerleri **sormak** için daha çok vaktiniz olur.
3. Ödevi(ve daha önemlisi kendinizi) **geliştirmek** için **daha çok vaktiniz** olur.



NOTLARIN DEĞERLENDİRİLMESİ VE DUYURULMASI

- Ödev süresi bitene kadar tekrar gönderim yapılabilir. **Son gönderim geçerlidir.**
- **Erken yapılan gönderimlere**, gönderim sırasına uyarak **geri bildirim yapılabilir.** Bu geri bildirimi alan öğrenci, ödev süresi dolmadıysa, kodunu daha da geliştirip tekrar gönderebilir.



Şekil 14 - Erken bitiren öğrencilere ödev süresi dolmadan geri bildirim sunulabilir.

NOTLANDIRMA ŞEKLİMİZ:

Bu ödevden oyununuzun algoritmik seviyesine göre temel puanınızı (40, 70, 100 ya da **100 üstü-maks. 200 puan**) alırsınız. Algoritmik seviyelerin tanımları için soru metnini inceleyiniz.

Kodunuz çalışıyorsa “hata” üretmiyor demektir. Ancak detaylandırılmış uyarı ayarları açıkken (bkz. SUNU01’in sonu) DevCpp’ta kodunuz “uyarı” üretiyor olabilir. (Kodu derlerken DevCpp’ta sağ üstte “...Release” seçili olmasına özen gösteriniz.) Kodunuzun ürettiği her uyarıdan **5 muhtemel puan kaybedersiniz.** Ayrıca kodunuz incelediğinde ortaya çıkacak hatalardan (örn. mantık hatalarından) da puan kaybedebilirsiniz.

Çok gelişmiş bir ödevle alabileceğiniz **maksimum 200 temel puan** ve erken gönderim ödülü olarak alabileceğiniz ekstra 15 puan ile birlikte ödevden 100 üzerinden 215 puan almak mümkündür.

LÜTFEN ÇÖZÜMLERİNİZİ İYİCE KONTROL EDİNİZ!



- **Detaylandırılmış uyarı ayarları** (bkz. SUNU01 sonu) açıkken ve sağ üstte "... Release" seçiliyken kodu DevCpp'ta çalıştırdığınızda uyarı almayacak şekilde kodunuzu güncellediniz mi?
- Kodunuz **rahat anlaşılır** (yorumlarla açık, girintilere dikkat edilmiş) bir kod mu?
- Kodunuzda da **"goto" ya da küresel değişken** kullanımı **olmadığına** emin misiniz?
- Kodunuzun **başında bir yorumla** bilgilendirme yaptınız mı?
- Kodunuzda **satırların yanına yorumlarla** işlevlerini açıkladınız mı?
- Kodunuzda her **süslü parantez kendi başına** bir satırda durarak ve **girintiler uygun şekilde** yapılmış olarak estetik açıdan güzel, rahat anlaşılır bir görünüm oluşturdu mu?
- Oyununuzun, **SYÇS6'daki sayı oyunu ya da taş-kağıt-makas oyunundan farklı** olmasına dikkat ettiniz mi?
- Kodunuz **mantıklı ve akıcı bir oyun** mudur? Oyuncunun anlayabileceği ve sıradan bir kullanıcının zevk alarak oynayabileceği bir kurguya sahip midir?
- Kodunuzda FONKSİYONLARI etkin şekilde kullanarak hem **geliştirilebilir** hem de **kolay anlaşılır** bir kod oluşturduğunuz mu?
- Derste **öğrendiğiniz** konuları kodunuza yansıtmaya çalıştınız mı?
- Kodunuzu değişik girdilerle dikkatlice **denediniz mi?**
- DevCpp'ın vereceği uyarı başına ödevinizden tek tek puan kırılabilceğini ve bunun için kodunuzu göndermeden önce **detaylandırılmış uyarı ayarları** açıkken kontrol etmiş olmanın önemini anlıyor musunuz?
- **Ödevlerdeki Etik İlkelerimiz'e** uyarak bu ödevi tamamladınız mı? Uymanızı engelleyen ya da aklınıza takılan bir durum olduysa bize danışınız. Gerekliyse (ödevin gecikmeli gönderim süresinin sonuna kadar) bize e-posta atarak ödevinizi geri çekebilirsiniz.