



DeepL

Subscribe to DeepL Pro to translate larger documents.  
Visit [www.DeepL.com/pro](https://www.DeepL.com/pro) for more information.



# Çok Uzun

## Ve tüm balıklar için teşekkürler!

*Özet:*

*Bu proje çok küçük bir 2D oyun.  
Amacı, dokular, sprite'lar ve diğer bazı çok temel oyun  
öğeleriyle çalışmanızı sağlamaktır.*

*Sürüm: 2.3*

# İçindekiler

<b>I</b>	<b>Önsöz</b>	<b>2</b>
<b>II</b>	<b>Hedefler</b>	<b>3</b>
<b>III</b>	<b>Ortak Talimatlar</b>	<b>4</b>
<b>IV</b>	<b>Zorunlu kısım</b>	<b>6</b>
IV.1	Oyun .....	7
IV.2	Grafik yönetimi .....	7
IV.3	Harita .....	8
<b>V</b>	<b>Bonus bölüm</b>	<b>9</b>
<b>VI</b>	<b>Örnekler</b>	<b>10</b>
<b>VII</b>	<b>Sunum ve akran değerlendirmesi</b>	<b>11</b>

# Bölüm I Önsöz

Geliştirici olmak, kendi oyununuzu yaratmak için harika bir şey.

Ancak iyi bir oyunun bazı iyi varlıklara ihtiyacı vardır. 2D oyunlar oluşturmak için karolar, karo setleri, sprite'lar ve sprite sayfaları aramanız gerekecektir.

Neyse ki, bazı yetenekli sanatçılar eserlerini [itch.io](https://itch.io) gibi platformlarda paylaşmaya istekli

Her durumda, diğer insanların çalışmalarına saygı göstermeye çalışın.

# Bölüm II Hedefler

Temel bir bilgisayar grafiği projesi oluşturmanın zamanı geldi!

Bu kadar uzun süre şu alanlarda becerilerinizi geliştirmenize yardımcı olacaktır: pencere yönetimi, olay işleme, renkler, dokular vb.

Okulun grafik kütüphanesini kullanacaksınız: MiniLibX! Bu kütüphane dahili olarak geliştirilmiştir ve bir pencere açmak, görüntüler oluşturmak ve klavye ve fare olaylarıyla başa çıkmak için gerekli temel araçları içerir.

Diğer hedefler bu ilk yıl için diğer tüm hedeflere benzer: titiz olmak, C programlamada seviye atlamak, temel algoritmaları kullanmak, biraz bilgi araştırması yapmak vb.

## Bölüm III

### Ortak Talimatlar

- Projeniz C dilinde yazılmış olmalıdır.
- Projeniz Norm'a uygun olarak yazılmış olmalıdır. Eğer bonus dosyalarınız/fonksiyonlarınız varsa bunlar norm kontrolüne dahil edilir ve içinde norm hatası varsa 0 alırsınız.
- Fonksiyonlarınız tanımlanmamış davranışlar dışında beklenmedik bir şekilde çıkmamalıdır (segmentasyon hatası, veri yolu hatası, çift serbest bırakma, vb. Böyle bir durumda, projeniz işlevsel değil olarak kabul edilecek ve değerlendirme sırasında 0 alacaktır.
- Yığın olarak tahsis edilen tüm bellek alanları gerektiğinde uygun şekilde serbest bırakılmalıdır. Hiçbir sızıntıya müsamaha gösterilmeyecektir.
- Konu gerektiriyorsa, kaynak dosyalarınızı -Wall, -Wextra ve -Werror bayraklarıyla gerekli çıktıya derleyecek bir Makefile sunmanız, cc kullanmanız ve Makefile'nızın yeniden bağlanmaması gerekir.
- Makefile dosyanız en azından \$(NAME), all, clean, fclean kurallarını içermeli ve re.
- Projenize bonuslar eklemek için, Makefile dosyanıza bir kural bonusu eklemelisiniz; bu bonus, projenin ana bölümünde yasak olan tüm çeşitli başlıkları, kütüphaneleri veya işlevleri ekleyecektir. Konu başka bir şey belirtmiyorsa bonuslar farklı bir \_bonus.{c/h} dosyasında olmalıdır. Zorunlu ve bonus kısım değerlendirmesi ayrı ayrı yapılır.
- Projeniz libft kullanmanıza izin veriyorsa, kaynaklarını ve ilişkili Makefile'ını ilişkili Makefile'ı ile birlikte bir libft klasörüne kopyalamanız gerekir. Projenizin Makefile'ı kütüphaneyi Makefile'ını kullanarak derlemeli ve ardından projeyi derlemelidir.
- Bu çalışma **teslim edilmek zorunda olmasa ve notlandırılmasa** bile projeniz için test programları oluşturmanızı teşvik ediyoruz. Bu size kendi çalışmanızı ve meslektaşlarınızın çalışmalarını kolayca test etme şansı verecektir. Bu testleri özellikle savunmanız sırasında faydalı bulacaksınız. Nitekim, savunma sırasında kendi testlerinizi ve/veya değerlendirdiğiniz akranınızın testlerini kullanmakta özgürsünüz.
- Çalışmanızı size atanmış git deposuna gönderin. Yalnızca git deposundaki çalışmalara not verilecektir. Çalışmanıza not vermek için Deepthought görevlendirilirse, bu işlem şu şekilde yapılacaktır



teşekkürler!

akran değerlendirmelerinizden sonra. Deepthought'un notlandırması sırasında çalışmanızın herhangi bir bölümünde bir hata meydana gelirse, değerlendirme duracaktır.





## Bölüm IV

### Zorunlu kısım

<b>Program adı</b>	so_long
<b>Dosyaları teslim edin</b>	Makefile, *.h, *.c, haritalar, dokular
<b>Makefile</b>	NAME, all, clean, fclean, re
<b>Argümanlar</b>	.ber biçiminde bir harita
<b>Harici fonksiyonlar.</b>	<ul style="list-style-type: none"><li>• open, close, read, write, malloc, free, perror, strerror, exit</li><li>• Matematik kütüphanesinin tüm fonksiyonları (-lm derleyici seçeneği, man man 3 math)</li><li>• MiniLibX'in tüm fonksiyonları</li><li>• ft_printf ve herhangi bir eşdeğer YOU kodlu</li></ul>
<b>Libft yetkili</b>	Evet
<b>Açıklama</b>	İçinde bir yunusun olduğu basit bir 2D oyun yaratmalısınız balık yedikten sonra Dünya'dan kaçır. Bir yunus, balık ve Dünya yer i n e , istediğiniz her h a n g i bir karakteri, herhangi bir koleksiyonu ve istediğiniz herhangi bir yeri kullanabilirsiniz.

Projeniz aşağıdaki kurallara uygun olmalıdır:

- MiniLibX kullanmalısınız. Ya okul makinelerinde bulunan sürümü ya da kaynaklarını kullanarak yükleyin.
- Kaynak dosyalarınızı derleyecek bir Makefile oluşturmalsınız. Yeniden bağlanmamalıdır.
- Programınız parametre olarak .ber ile biten bir harita açıklama dosyası almalıdır uzatma.

## IV.1 Oyun

- Oyuncunun amacı, haritada bulunan tüm koleksiyon malzemelerini toplamak ve ardından mümkün olan en kısa yolu seçerek kaçmaktır.
- Ana karakteri hareket ettirmek için W, A, S ve D tuşları kullanılmalıdır.
- Oyuncu şu **4 yönde** hareket edebilmelidir: yukarı, aşağı, sol, sağ.
- Oyuncu duvarlara doğru hareket edememelidir.
- Her hamlede, mevcut **hareket sayısı** kabukta görüntülenmelidir.
- **2B görünüm** (yukarıdan aşağıya veya profil) kullanmanız gerekir.
- Oyun gerçek zamanlı olmak zorunda değil.
- Verilen örnekler yunus temasını gösterse de siz istediğiniz dünyayı yaratabilirsiniz.



İsterseniz ana karakterinizi hareket ettirmek için ZQSD'yi veya klavyenizdeki ok tuşlarını kullanabilirsiniz.

## IV.2 Grafik yönetimi

- Programınızın görüntüyü bir pencerede göstermesi gerekir.
- Pencerenizin yönetimi sorunsuz kalmalıdır (başka bir pencereye geçme, simge durumuna küçültme vb.).
- ESC tuşuna basıldığında pencere kapanmalı ve programdan temiz bir şekilde çıkılmalıdır.
- Pencerenin çerçevesindeki çarpı işaretine tıklandığında pencere kapanmalı ve programdan temiz bir şekilde çıkılmalıdır.
- MiniLibX'in görüntülerinin kullanılması zorunludur.

### IV.3 Harita

- Harita 3 bileşenle inşa edilmelidir: **duvarlar**, **toplanabilir eşyalar** ve **boş alan**.
- Harita sadece bu 5 karakterden oluşabilir:  
**0** boş bir alan için,  
**1** bir duvar için,  
**C** Koleksiyon için,  
**E** Harita çıkışı için,  
**P** oyuncunun başlangıç pozisyonu

için. İşte basit bir geçerli harita:

```
11111111111111
100100000000C1
1000011111001
1P0011E000001
11111111111111
```

- Haritanın geçerli olması için **1 çıkış**, en az **1 toplanabilir** ve **1 başlangıç konumu** içermesi gerekir.



Harita yinelenen karakterler (çıkış/başlangıç) içeriyorsa, bir hata mesajı görüntüleniz gerekir.

- Harita dikdörtgen şeklinde olmalıdır.
- Harita kapalı/duvarlarla çevrili olmalıdır. Eğer değilse, program bir hata döndürmelidir.
- Haritada geçerli bir yol olup olmadığını kontrol etmelisiniz.
- Yukarıdaki kurallara uyduğu sürece her türlü haritayı ayrıştırabilmelisiniz.
- Minimal .ber haritasına bir başka örnek:

[illegible]

- Dosyada herhangi bir yanlış yapılandırmayla karşılaşırsa, program temiz bir şekilde çıkmalı ve "Error\n" ve ardından seçtiğiniz açık bir hata mesajını döndürmelidir.

# Bölüm V Bonus

## kısım

Genellikle, kendi orijinal ekstra özelliklerinizi geliştirmeye teşvik edilirsiniz. Ancak, daha sonra çok daha ilginç grafik projeleri olacak. Onlar sizi bekliyor!!! Bu ödev için çok fazla zaman kaybetmeyin!

Değerlendirmeniz sırasında kullanımları **gerekçelendirildiği** sürece bonus bölümünü tamamlamak için diğer işlevleri kullanmanıza izin verilir. Akıllı olun!

Eğer yaparsanız ekstra puan alacaksınız:

- Oyuncu bir düşman devriyesine dokunduğunda kaybetsin.
- Biraz sprite animasyonu ekleyin.
- Hareket sayısını kabuğa yazmak yerine doğrudan ekranda görüntüleyin.



Gerektiğinde bonuslara göre dosya/klasör ekleyebilirsiniz.



Bonus bölüm sadece zorunlu bölümün MÜKEMMEL olması halinde değerlendirilecektir. Mükemmel, zorunlu kısmın bütünsel olarak yapıldığı ve hatasız çalıştığı anlamına gelir. TÜM zorunlu gereklilikleri geçmediyseniz, bonus bölümünüz hiç değerlendirilmeyecektir.

# Bölüm VI

## Örnekler



Grafik tasarımda berbat bir zevki gösteren çok uzun örnekler  
(neredeyse bazı bonus puanlara değer)!

## **Bölüm VII**

### **Sunum ve akran değerlendirmesi**

Ödevinizi her zamanki gibi Git deponuzda teslim edin. Savunma sırasında yalnızca deponuzdaki çalışmalar değerlendirilecektir. Doğru olduklarından emin olmak için dosyalarınızın adlarını iki kez kontrol etmekten çekinmeyin.

Bu ödevler bir program tarafından doğrulanmadığından, zorunlu dosyaları teslim ettiğiniz ve gerekliliklere uyduğunuz sürece dosyalarınızı istediğiniz gibi düzenlemekten çekinmeyin.