



SoLong

Ve tüm balıklar için teşekkürler!

Özet: Bu proje çok küçük bir 2D oyundur. Sizi doku ve hareketli grafiklerle çalıştırmak için hazırlandı. Ve bazı çok temel oynanış elemanları..

Versiyon: 1

İçindekiler

I	Önsöz	2
II	Hedefler	3
III	Genel Talimatlar	4
IV	Zorunlu kısım - so long	6
V	Bonus bölüm	9
VI	Örnekler	10

Bölüm I

Önsöz

Geliştirici olmak kendi oyununuzu yaratmak için harika bir şeydir.

Ama iyi bir oyun bazı iyi assetlere ihtiyaç duyar.

2D oyunlar için tiles, tilesets, hareketli grafikler, ve hareketli grafik sayfalarını araştırmalısınız.

Ne güzel ki, bazı yetenekli sanatçılar kendi çalışmalarını itch.io gibi platformlarda paylaşmak konusunda hevesliler.

Ne yaparsanız yapın diğerlerinin çalışmasına saygı duyun.

Bölüm II

Hedefler

Bu projenin amaçları bu senenin diğer projeleri ile benzer: Titizlik, C kullanımı, temel algoritmaların kullanımı, araştırma yapmak vb.

Bir grafik tasarım projesi olarak, **so long** sizin şu alandaki yeteneklerinizi geliştirmenize olanak sağlayacaktır: pencereler, renkler, olaylar, dokular vb.

Bölüm III

Genel Talimatlar

- Projeleriniz C programlama dilinde yazılmalıdır.
- Projeleriniz Norm'a uygun olarak yazılmalıdır. Bonus dosyalarınız/fonksiyonlarınız varsa, bunlar norm kontrolüne dahil edilir ve bu dosyalarda norm hatası varsa 0 alırsınız.
- Tanımlanmamış davranışlar dışında sizin fonksiyonlarınız beklenmedik bir şekilde sonlanmamalıdır (Segmentasyon hatası, bus hatası, double free hatası, vb.) . Eğer bunlar yaşanır s 0 alırsınız.
- Heap'de ayırmış olduğunuz hafıza adresleri gerekli olduğu durumlarda serbest bırakılmalıdır. Hiçbir istisna tolere edilmeyecektir.
- Eğer verilen görev **Makefile** dosyasının yüklenmesini istiyorsa, sizin kaynak dosyalarınızı **-Wall**, **-Wextra** , **-Werror**, flaglarını kullanarak derleyip çıktı dosyalarını üretecek olan **Makefile** dosyasını oluşturmanız gerekmektedir. **Makefile** dosyasını oluştururken **cc** kullanın ve **Makefile** dosyanız yeniden ilişkilendirme yapmamalıdır (re-link).
- **Makefile** dosyanız en azından **\$(NAME)**, **all**, **clean**, **fclean** ve **re** kurallarını içermelidir.
- Projenize bonusu dahil etmek için **Makefile** dosyanıza **bonus** kuralını dahil etmeniz gerekmektedir. Bonus kuralının dahil edilmesi bu projenin ana kısmında kullanılması yasak olan bazı header dosyaları, kütüphaneler ve fonksiyonların eklenmesini sağlayacaktır. Eğer projede farklı bir tanımlama yapılmamışsa, bonus projeleri **_bonus.{c/h}** dosyaları içerisinde olmalıdır. Ana proje ve bonus proje değerlendirmeleri ayrı ayrı gerçekleştirilmektedir.
- Eğer projeniz kendi yazmış olduğunuz **libft** kütüphanesini kullanmanıza izin veriyorsa, bu kütüphane ve ilişkili **Makefile** dosyasını proje dizinindeki **libft** klasörüne ilişkili **Makefile** dosyası ile kopyalamanız gerekmektedir. Projenizin **Makefile** dosyası öncelikle **libft** kütüphanesini kütüphanenin **Makefile** dosyasını kullanarak derlemeli ardından projeyi derlemelidir.
- Test programları sisteme yüklenmek zorunda değildir ve puanlandırılmayacaktır. Buna rağmen test programları yazmanızı şiddetle önermekteyiz. Test programları

sayesinde kendinizin ve arkadaşlarınız projelerinin çıktılarını kolaylıkla gözlemleyebilirsiniz. Bu test dosyalarından özellikle savunma sürecinde çok faydalanacaksınız. Savunma sürecinde kendi projeleriniz ve arkadaşlarınızın projeleri için test programlarını kullanmakta özgürsünüz.

- Çalışmalarınız atanmış olan git repolarına yüklemeniz gerekmektedir. Sadece git reposu içerisindeki çalışmalar notlandırılacaktır. Eğer Deepthought sizin çalışmanızı değerlendirmek için atanmışsa, bu değerlendirmeyi arkadaşlarınızın sizin projenizi değerlendirmesinden sonra gerçekleştirecektir. Eğer Deepthought değerlendirme sürecinde herhangi bir hata ile karşılaşılırsa değerlendirme durdurulacaktır.

Bölüm IV

Zorunlu kısım - so long

Program adı	so_long
Teslim edilecek dosyalar	Tüm dosyalarınız
Makefile	all, clean, fclean, re, bonus
Argümanlar	*.ber formatında bir harita
Harici fonksiyonlar.	<ul style="list-style-type: none">• open, close, read, write, printf, malloc, free, perror, strerror, exit• MinilibX ' in tüm fonksiyonları
Libft kullanılabilir mi?	Evet
Açıklama	Biraz balık yedikten sonra ortamdan kaçan bir yunusun olduğu 2D bir oyun oluşturmalısınız. Veya tüm kahramanlar mekanı terketmeden önce tüm değerli eşyaları toplasın.

Kısıtlamalar şu şekildedir:

- miniLibX kullanmak zorundasınız. İşletim sisteminde olanı veya kendi kaynaklarından aldığınızı kullanabilirsiniz. Eğer kaynakları tercih ederseniz, libft ile aynı kuralları Ortak Talimatlar bölümünde yazdığı şekilde uygulamalısınız.
- Penceredeki işlemler takılmadan (pürüzsüzce) olmalıdır. Başka bir pencereye geçmek, pencereyi küçültmek vb.
- Örnekler yunus temasıyla verilmiş olsa da siz istediğinizi kullanabilirsiniz.

- Harita 3 bileşenden oluşacaktır: Duvarlar, toplanabilirler ve boş alan.
 - Oyuncunun hedefi haritada sunulan tüm toplanabilirleri toplamak ve en az hareketle kaçmaktır.
 - Her hareketten sonra güncel hareket sayısı shell'de bastırılmalıdır.
 - Oyuncu hareket edebilmelidir: yukarı, aşağı, sol, sağ.
 - 2D görüş kullanacaksınız. (yukarıdan aşağıya veya profilden).
 - Oyun gerçek zamanlı olmak zorunda değildir.
 - Oyuncu duvarın içinde hareket edemez.
 - Program ekrana görüntüyü gösterecek ve aşağıdaki kuralları takip edecektir:
 - W, A, S, ve D tuşları ana karakteri hareket ettirmek için kullanılacaktır
 - ESC'ye basmak pencereyi kapatmalı ve programdan temizce çıkmalıdır.
 - Pencerenin çerçevesindeki kırmızı çarpı işaretine basıldığında pencere kapanmalı ve programdan temizce çıkılmalıdır.
 - minilibX'deki görselin kullanılması şiddetle önerilmektedir.
 - Program harita tanımını .ber uzantısıyla beraber argüman olarak almalıdır.
 - Harita sadece 5 olası karakterden oluşmalıdır: Boş alan için 0, duvar için 1, toplanabilirler için C, haritadan çıkış için E ve oyuncunun başlangıç konumu için P.
- This is a simple valid map:
- ```
11111111111111
10010000000C1
1000011111001
1P0011E000001
11111111111111
```
- Harita duvarlar tarafından kapanmalı/çevrelenmelidir. Eğer öyle değilse program hata vermelidir.
  - Harita en az bir çıkış, bir toplanabilir, ve bir başlangıç pozisyonu içermelidir.
  - Haritada doğru bir yol var mı diye kontrol etmenize gerek yok.
  - Harita dikdörtgen şeklinde olmalıdır.
  - Harita kurallarına uyduğu sürece her türlü haritayı ayrıştırıp incelemelisiniz.



- Bir diğer küçük **.ber** haritası:

```
111
1E00000000000000C000000C0000000000000000000000000000000
1010010100100000101001000000010101
1010010010101010001001000000010101
1P0000000C00C00000000000000000000000000000000000000C1
111
```

- Eğer dosyada herhangi bir yanlışlık varsa program hemen kapanmalı ve sizin seçiminize göre açıklayıcı bir hata mesajını "Error\n" bastıktan sonra basmalıdır

## Bölüm V

### Bonus bölüm



Bonus kısım, yalnızca zorunlu kısım MÜKEMMEL ise değerlendirilecektir. Mükemmel, zorunlu kısmın entegre olarak yapıldığı ve arızasız çalıştığı anlamına gelir. TÜM zorunlu gereksinimleri geçmediyseniz, bonus bölümünüz hiç değerlendirilmeyecektir.

Değerlendirme sırasında sıkıntı çıkarmayacağı sürece bonus kısımda istediğiniz fonksiyonları kullanabilirsiniz. Akıllı hareket edin!

Bonus listesi:

- Temas halinde oyuncunun kaybetmesine neden olacak düşman hareketli düşman birimleri.
- Bazı hareketli animasyonlar.
- Hareket sayısının shell yerine direct olarak ekranda bastırılması.



İleride daha iyi oyunlar yapabileceksiniz. Zamanınızı çok fazla boşa harcamayın!



# Bölüm VI

## Örnekler

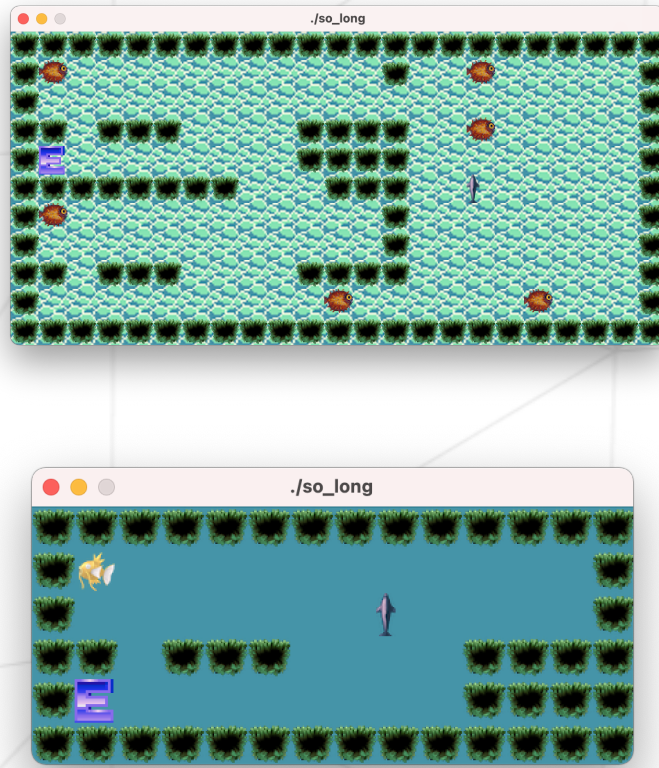


Figure VI.1: : Pek tat vermeyen (neredeyse bonus hak edecek kadar) bazı minilibX örnekleri!