# KOCAELİ ÜNİVERSİTESİ
# BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ
# PROGRAMLAMA LABORATUVARI II - 1. Proje

---

**PROJE TESLİM TARİHİ: 28.03.2021**

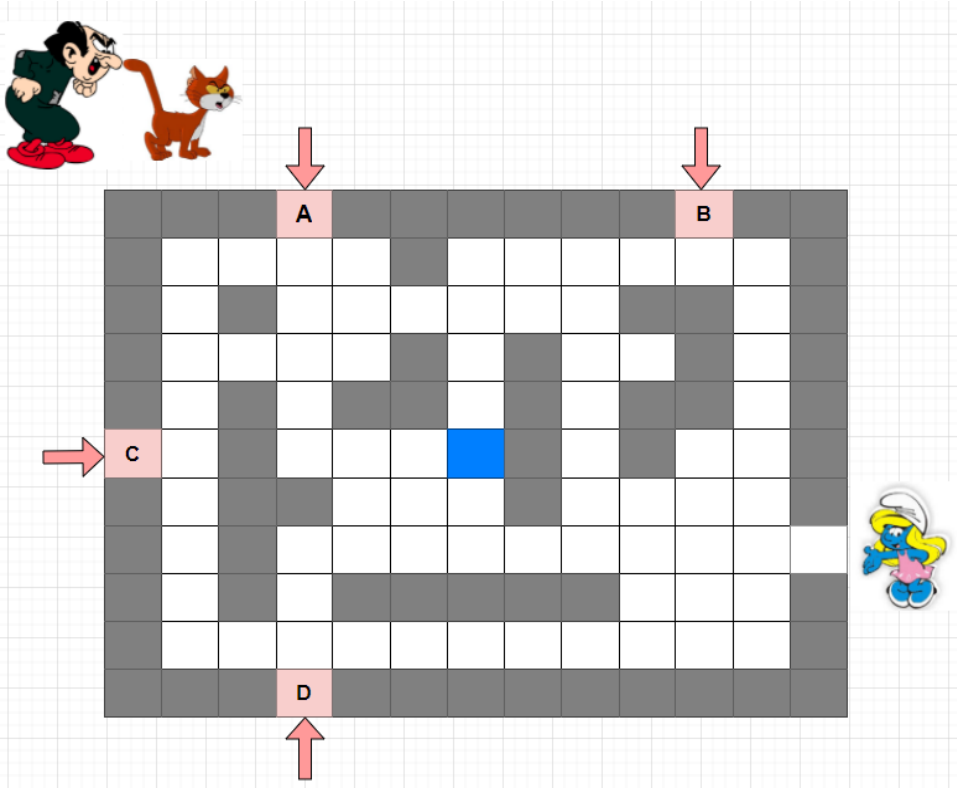
# ŞİRİNLER


## Projenin Amacı:

Bu proje ile nesneye yönelik programlama ve veri yapıları algoritmalarını kullanarak Şirinler oyunu tasarlamanız beklenmektedir. Seçilen oyuncunun Labirent içerisinde puanlarını bitirmeden önce Şirine'ye ulaşması gerekmektedir.

Hedefe doğru giderken oyuncuyu birçok zorluk beklemektedir. Oyun içerisinde iki farklı oyuncu ve iki farklı düşman karakterleriniz olacaktır. Kullanıcı oyun başlamadan önce seçtiği oyunculardan birini klavye yardımı ile kontrol edecek ve Şirine'ye ulaştırmaya çalışacaktır. Yine oyun başlamadan önce seçmiş olduğu düşman karakterlerden biri veya birden fazlası da onu durdurmaya çalışacaktır. Şekil 1'de, projede kullanacağınız harita verilmiştir. Mavi renkli kutu

oyuncuların (yani kullanıcı tarafından kontrol edilecek olan karakter) başlama noktası olacaktır. Kırmızı oklar ise düşman karakterlerin labirente giriş yapabileceği kapılar olacaktır. Kullanıcının kontrol ettiği oyuncu puanları bitmeden önce Prensese ulaşmalıdır aksi halde oyuncu başarısız olur.




Şekil 1: Projede kullanılacak harita

**Programlama Dili:** Proje **C++ veya Java** dili kullanılarak gerçekleştirilecektir.

**En kısa yol algoritması**: Sadece **Dijisktra algoritması** kullanılacaktır.

## 1. İsterler

Sizden bir arayüz tasarlamanız beklenmektedir. Şekil 1'de verilen haritayı oluşturmak için grafik kütüphanelerinden yararlanabilirsiniz. Bunun için graphics.h veya allegro.h gibi kütüphanelerinden faydalanabilirsiniz (Grafik için kütüphane sınırlaması yoktur). Başlangıçta seçilen oyuncu 20 Puanla oyuna başlayacak ve oyuncu puanı 0 ya da 0'ın altına düştüğünde oyun oyuncunun başarısızlığı ile sonlanacaktır.

**Karakterler:**

**Oyuncular:** Kullanıcıdan oyun başlamadan önce iki farklı oyuncudan birini seçmesi beklenmektedir. Bu oyunculardan biri **Gözlüklü Şirin** diğeri ise **Tembel Şirin**'dir. Gözlüklü

Şirin'nin Tembel Şirin'e göre farklı bir özelliği bulunmaktadır. Gözlüklü Şirin, düşman karaktere dokunduğu anda puanı farklı şekilde azalacaktır.

**Düşman karakterler: Gargamel ve Azman** isimli iki farklı düşman karakter bulunmaktadır. Gargamel, Azman'a göre daha hızlıdır. Tek bir harekette iki birim birden ilerlemektedir. Azman ise her hamlede tek bir birim ilerler ve hiçbir ek özelliği bulunmamaktadır.

**Şekil 2. Karakterler**

## 2. Projenin Veri Yapıları ile İlgili Kısmı

Kullanıcının seçmiş olduğu oyunculardan biri Şirine'ye ulaşmaya çalışırken;

- Düşman karakterlerden bir veya daha fazlası onu engellemeye çalışacaktır.
- Bu esnada oyuncuların hareketlerine bağlı olarak dinamik **en kısa yolu** bularak gerçekleştirecektir.
- **Bu en kısa yolu da harita üzerinde çizdirerek göstermesi gerekmektedir.**
- Bunun için farklı algoritma kalıplarından yararlanabilirsiniz, yararlandığınız algoritmayı zaman ve bellek karmaşıklığı yönünden analiz etmeniz gerekmektedir.

**Projenin Nesneye Yönelik Programlama ile İlgili Kısmı**

- Projede oyuncular ve düşman karakterin ortak özellikleri bulunmakta burada sizden beklenen hiyerarşik bir sınıf mimarisi kurmanızdır.

- Karakter sınıfı en temel sınıfınız olacak, oyuncular ve düşman karakter bu sınıflardan türeyecek. **Oyuncuların en kısa yol hesabı yapma işlemi olmamasına rağmen düşman karakterde en kısa yol hesabı yer almaktadır.**

## 3. İstenen Sınıflar

### Karakter Sınıfı

**Bu sınıfta bulunması gereken özellikler ve fonksiyonlar:**

- Karakterlerin adını tutacak ID, Ad, Türünü (oyuncu/düşman) tutacak.
- Karakterlerin ilerlediği koordinatları tutacak Lokasyon değişkenleri olmalıdır.
- Constructor, Get, Set ve En KısaYol metotları yer almalıdır.

### Oyuncu Sınıfı

Bu sınıfta **karakter sınıfı** kalıtım olarak verilecektir. Constructor, Get ve Set metotları yer alacaktır. Oyuncunun düşman karaktere her dokunuşunda kazandığı puan kademeli olarak düşecektir. Bu sınıftaki parametreler oyuncuID, oyuncuAdi, oyuncuTur ve ayrıca Skor şeklinde olmalıdır. PuaniGoster() fonksiyonu ile oyuncuların skorları gösterilecektir.

**Oyuncu Türlerine Alt Sınıflar (3 Adet):**

- **Gözlüklü şirin**(Oyuncu 1): Her defasında 2 birim ilerler. Azman'a dokunursa oyuncu 5 puan kaybeder. Gargamel'e dokunursa oyuncu 15 puan kaybeder.
- **Tembel Şirin**(Oyuncu 2): Her defasında 1 birim ilerler. Azman'a dokunursa oyuncu 5 puan kaybeder. Gargamel'a dokunursa oyuncu 15 puan kaybeder.
- **Puan** : Oyuncu sınıfında bulunan PuaniGoster() metotu override edilerek her bir oyuncu için özelleştirilecektir.

**Not**: Hiçbir oyuncu çapraz gidemez sadece sağ, sol, yukarı ya da aşağı yönde hareket sağlayabilir.

### Düşman Sınıfı

Bu sınıfta **karakter sınıfı** kalıtım olarak verilecektir. Constructor, Get ve Set metotları yer alacaktır. Bu sınıftaki Parametreler dusmanID, dusmanAdi ve dusmanTür şeklinde olmalıdır.

**Düşman Türlerine Alt Sınıflar (2 Adet):**

- Azman (Düşman 1): 1 birim ilerler. Azman Gargamel'in üzerinden atlayamaz.
- Gargamel (Düşman 2): 2 birim ilerler. Gargamel Azman'nın üzerinden atlayabilir.

**Lokasyon sınıfı**

x ve y koordinatlarını tutan iki farklı değişken tutulmalı. Constructor, Get ve Set metotları yer almalıdır.

## Obje Sınıfı

**Obje Türlerine Alt Sınıflar (2 Adet):**

- Altın : Her bir altın 5 puan değerindedir. Ek olarak altın sayısını tutmak için altın değişkeni tutulacak. Random olarak (maksimum 10 saniye olacak şekilde) herhangi bir yerde aynı anda **5 altın oluşacak** (farklı sayıda altın oluşmayacaktır) ve 5 sn ekranda görünecektir. Eğer Oyuncu altını alabilirse her bir altın için 5 puan kazanır. Oyuncu altını alamazsa altın kaybolur.

- Mantar: Random olarak (maksimum 20 saniye olacak şekilde) herhangi bir yerde mantar oluşacak ve 7 sn ekranda görünecektir. Eğer Oyuncu mantarı alabilirse 50 puan kazanır. Oyuncu mantarı alamazsa mantar kaybolur.

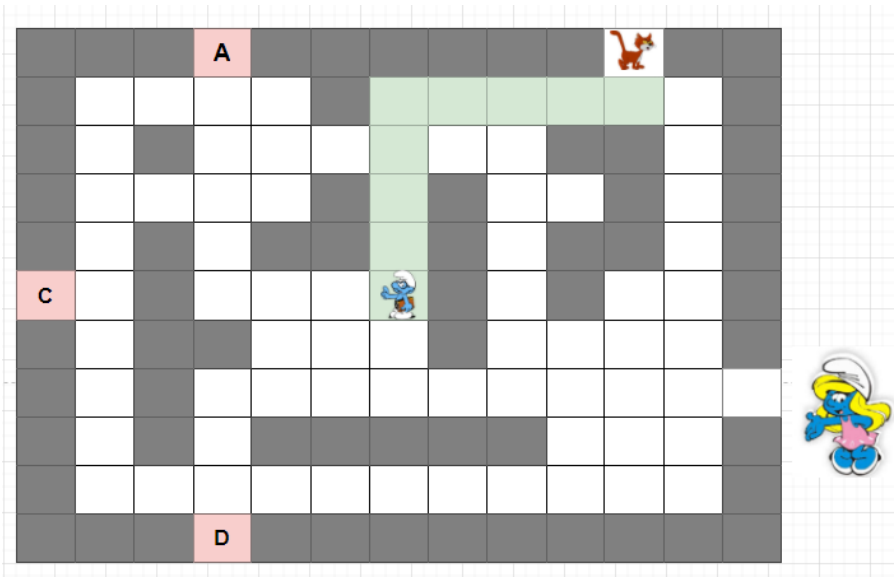
**Oyun Adımları**

- Program; harita ve oyuncu bilgilerini tek bir txt dosyasından okuyacaktır. Harita.txt dosyasında hangi düşman karakterin oyunda olacağı ve hangi kapıdan giriş yapacağı bilgisi bulunmalıdır. Size verilen harita sabit kalacak fakat karakter bilgilerinin değiştirilebilir olması beklenmektedir bunun için dinamik yapı oluşturmalısınız. List, queue, stack veya dizi kullanabilirsiniz.

- Program çalıştırıldığında kullanıcıya harita grafiksel olarak sunulmalıdır

- Sol üst köşenin (0,0) noktası olduğunu unutmayınız bu durumda oyuncunun başlayacağı sarı nokta (5,6) noktası olarak belirlenmiştir. **Mavi karede** seçtiğiniz Gözlüklü Şirin ya da Tembel Şirin oyuncusu bulunacaktır. Bunun için kullanacağınız görselleştirme işlemi size bırakılmıştır fakat hangi karakterin oynadığı belirli olmalıdır.

- **Program başlar başlamaz, düşman karakter oyuncuya en kısa kaç adımda ulaşabileceği bilgisini her adımda vermelidir**. Örneğin Gözlüklü Şirin kullanıcı tarafından seçilen oyuncu ve Azman da düşman karakter olsun (birden fazla olabilir) Azman'ın B kapısından giriş yaptığı varsayılsın bu durumda sizden yapılmasını beklenen Şekil 3'teki gibi bir görüntü olacaktır.




**Şekil 3. Azman (0,10) 'dan (5,6)'ya 9 adımda ulaşmaktadır.**

- Oyuncu kontrolü kullanıcıda olacak ve klavye (oklar yardımı ile) üzerinden yapılacaktır.
- Oyuncu hareket ettikten sonra düşman karakter hareket edecektir. **En kısa yol her adımda harita üzerinde çizdirilecektir.**
- Eğer oyuncu ve düşman karakter birbirlerine dokunursa, oyuncunun puanı azaltılacaktır ve düşman karakter çıktığı kapıya geri dönecek ve oyun burdan devam edecektir.

**Not:** Harita.txt yi açmak için NotePad++ kullanınız.

**Her sınıf için ortak olan özellikler:**

- **Projede Encapsulation, Inheritance, Polymorphism, Abstraction yapılarının kullanılması beklenmektedir.**
- **Yapıcı (constuctor) metotları (parametreli ve parametresiz olarak en az iki) yazılacaktır.**
- **Tüm özellikler için get, set metotları tanımlanacaktır.**