

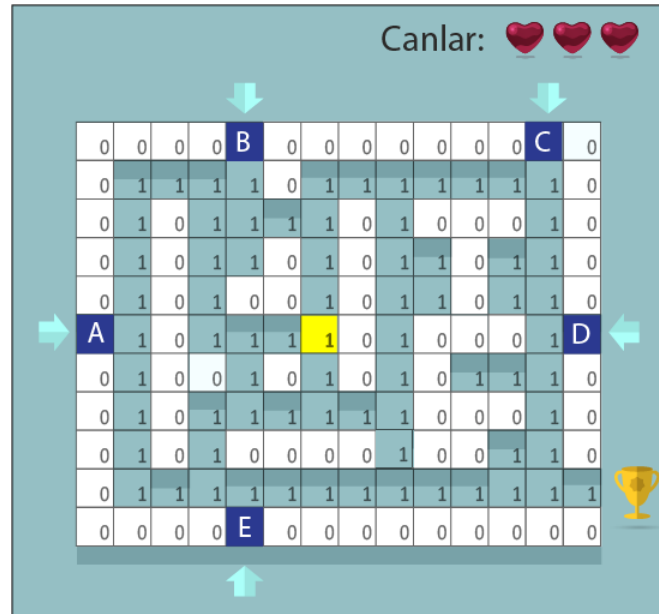
KOCAELİ ÜNİVERSİTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ
BLM210 PROGRAMLAMA LAB. II, 2018-2019
PROJE 1

PROJE TESLİM TARİHİ: 17.03.2019

STAR WARS

Bu projenin amacı, veri yapıları ve nesneye yönelik konularını nasıl uygulayacağınızı göstermektir.

Problem: Labirent içerisinde canlarınızı bitirmeden önce hedefe ulaşmanız gerekmektedir. Hedefe doğru giderken birçok zorluk sizi beklemekte olacak. Oyun içerisinde iki farklı kötü ve iki farklı iyi karakterleriniz olacak. Kullanıcı oyun başlamadan önce seçtiği iyi karakterlerden birini klavye yardımı ile kontrol edecek ve hedefe ulaşmaya çalışacak, yine oyun başlamadan önce seçmiş olduğu kötü karakterlerden biri veya birden fazlası da onu durdurmaya çalışacak. Şekil 1, projede kullanacağımız harita olacak. Sarı renkli kutu iyi karakterin (yani kullanıcı tarafından kontrol edilecek olan karakter) başlama noktası olacaktır. Mavi oklar ise kötü karakterin labirente giriş yapabileceği kapılar olacaktır. Kullanıcının kontrol ettiği iyi karakter canları bitmeden önce kupaya ulaşmalıdır aksi halde 'Game Over'.



Şekil 1. Harita

1. Karakterler

İyi karakterler: Kullanıcıdan oyun başlamadan önce iki farklı iyi karakterden birini seçmesi beklenmektedir. Bu karakterlerden biri Luke Skywalker diğeri ise Master Yoda olacak. Master Yoda'nın Luke Skywalker'a göre farklı bir özelliği bulunmakta. Master Yoda, kötü karakter tarafından yakalandığı anda canının sadece yarısı azalacaktır. Toplamda 3 can olduğu göz önüne alındığında 'GAME OVER' olması için Luke Skywalker'ın 3 kere, Master Yoda'nın da 6 kere yakalanması gerekmektedir.

Kötü karakterler: Darth Vader, Stormtrooper ve Kylo Ren olarak üç farklı kötü karakter bulunmaktadır. Darth Vader labirent içerisinde tüm duvarları yıkabilmektedir. Kylo Ren ise hızlıdır. Tek bir harekette iki basamak birden ilerlemektedir. Stormtrooper her hamlede tek bir birim ilerler ve hiçbir ek özelliği bulunmamaktadır.



Şekil 2. Karakterler

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

2. İsterler

Bir arayüz tasarlamamız beklenmektedir. Şekil 1'de verilen harita kullanılacak, bunun için grafik kütüphanelerinden yararlanabilirsiniz. Bunun için graphics.h veya allegro.h gibi kütüphanelerinden faydalanabilirsiniz (Grafik için kütüphane sınırlaması yoktur).

Projenin Veri Yapıları ile İlgili Kısım

Kullanıcı seçmiş olduğu iyi karakterden biri ile kupaya ulaşmaya çalışırken. Kötü karakterlerden bir veya birden fazlası da onu kovalayacaktır. Kovalama esnasında iyi karakterin hareketine bağlı olarak dinamik **en kısa yolu** olarak gerçekleştirecektir bu işlemi. Bunun için farklı algoritma kalıplarından yararlanabilirsiniz, yaralandığınız algoritmayı zaman ve bellek karmaşıklığı yönünden analiz etmeniz gerekmektedir.

Projenin Nesneye Yönelik Programlama ile İlgili Kısım

Projede iyi karakter ve kötü karakterin ortak özellikleri bulunmakta burada sizden beklenen hiyerarşik bir sınıf mimarisi kurmanız. Karakter sınıfı en temel sınıfınız olacak, iyi karakter ve kötü karakter bu sınıflardan türeyecek. İyi karakterin en kısa yol hesabı yapma işlemi olmamasına rağmen kötü karakterde bu işlemler yer almaktadır.

Aşağıda yer alan sınıfları uygulamanız beklenmektedir.

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

Karakter sınıfı: Karakterin adını tutacak Ad, turunu (iyi/kötü) tutacak Tur, ve karakterin ilerlediği koordinatları tutacak Lokasyon değişkenleri olmalıdır. Constructor, Get, Set ve EnKısaYol metotları yer almalıdır.

DarthVader sınıfı: Karakter sınıfı kalıtım olarak verilecek. Constructor, Get, Set ve EnKısaYol metotları yer almalıdır. EnKısaYol, tüm matris elemanlarının bir olacağı göz önüne alınarak hesaplanmalı.

KyloRen sınıfı: Karakter sınıfı kalıtım olarak verilecek. Constructor, Get, Set ve EnKısaYol metotları yer almalıdır. EnKısaYol, tek harekette iki birim birden gideceği göz önüne alınarak hesaplanmalı.

Stormtrooper sınıfı: Karakter sınıfı kalıtım olarak verilecek. Constructor, Get, Set ve EnKısaYol metotları yer almalıdır.

MasterYoda sınıfı: Karakter sınıfı kalıtım olarak verilecek. Constructor, Get ve Set metotları yer alacak. Ek olarak Can sayısını tutmak için can değişkeni yer alacak. Her yakalanışında canının sadece %50 si gidecek örneğin ilk kez yakalanma durumunda şekil 2’de yer alan durum oluşmalı.



Şekil 3. Can Yönetimi

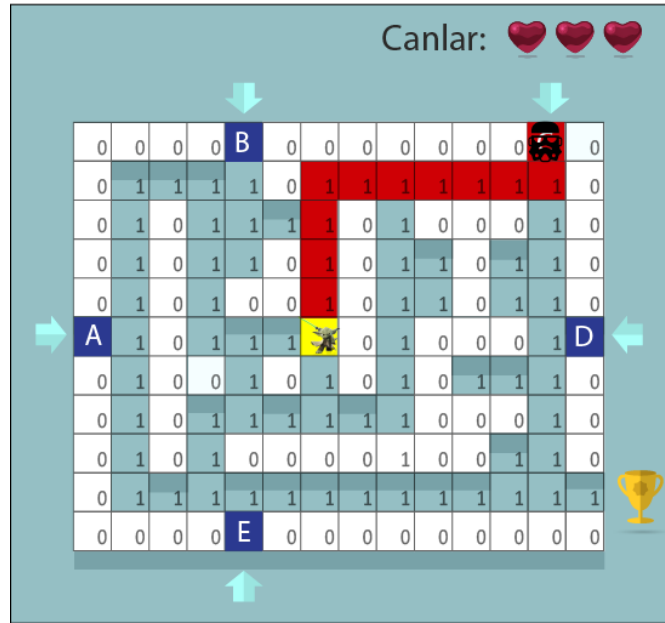
LukeSkywalker sınıfı: Karakter sınıfı kalıtım olarak verilecek. Constructor, Get ve Set metotları yer alacak. Ek olarak Can sayısını tutmak için can değişkeni yer alacak.

İstenilenler:

- Program, harita ve karakter bilgilerini tek bir txt dosyasından okuyacaktır. Harita.txt dosyasında hangi kötü karakterin oyunda olacağı ve hangi kapıdan giriş yapacağı

bilgisi bulunmalı. 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 sunulmalı.
- Sol üst köşenin (0,0) noktası olduğunu unutmayınız bu durumda oyuncunun başlayacağı sarı nokta (5,6) noktası olarak belirlenmiştir. **Sarı karede** seçtiğiniz Master Yoda ya da Luke Skywalker karakteri yer alacak. bunun için kullanacağınız görselleştirme işlemi size bırakılmıştır fakat hangi karakterin oynadığı belirli olmalı.
- Program başlar başlamaz, kötü karakter iyi karaktere en yakın kaç adımda ulaşabileceği bilgisini her adımda vermelidir. Örneğin Master Yoda kullanıcı tarafından seçilen iyi karakter ve Stormtrooper da kötü karakter olsun (birden fazla olabilir) Stormtrooper'un C kapısından giriş yaptığı varsayılın bu durumda sizden yapılmasını beklenen Şekil 4'ki gibi bir görüntü olacaktır.



Not: Şekil 1'de yapılan düzeltme burada da geçerlidir. Sarı kare 5. Satır (Sol köşe noktanın 0,0 olduğunu unutmayınız) 6. Sütunda yer almaktadır.

- İyi karakter kontrolü kullanıcıda olacak ve klavye (oklar yardımı ile) üzerinden yapılacak.
- İyi karakter hareket ettikten sonra kötü karakter hareket edecek. En kısa yol her adımda harita üzerinde çizdirilecek.
- Eğer oyuncu yakalanırsa, can azaltılacak ve oyun yeniden başlatılacak.

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

3. Ödev Teslimi

- Proje sunum gününde rapor (hard copy) teslim edilmesi gerekmektedir.

- Rapor ieee formatında (önceki yıllarda verilen formatta) 4 sayfa, akış diyagramı veya yalancı kod içeren, özet, giriş, yöntem, deneysel sonuçlar, sonuç ve kaynakça bölümünden oluşmalıdır.
- Dersin takibi projenin teslimi dahil edestek.kocaeli.edu.tr sistemi üzerinden yapılacaktır. edestek.kocaeli.edu.tr sitesinde belirtilen tarihten sonra getirilen projeler kabul edilmeyecektir.
- Proje ile ilgili sorular edestek.kocaeli.edu.tr sitesindeki forum üzerinden Arş. Gör. Seda Kul veya Arş. Gör. Fulya Akdeniz'e sorulabilir.
- Demo tarihleri daha sonra duyurulacaktır.
- Demo sırasında algoritma, geliştirdiğiniz kodun çeşitli kısımlarının ne amaçla yazıldığı ve geliştirme ortamı hakkında sorular sorulabilir.
- Kullandığınız herhangi bir satır kodu açıklamanız istenebilir.

KOPYA ÇEKTİĞİ YA DA KOPYA VERDİĞİ TESPİT EDİLEN ÖĞRENCİLER İÇİN KOPYA İŞLEMİ UYGULANACAKTIR.

PROJE EN FAZLA İKİ KİŞİLİK GRUPLAR HALİNDE YAPILACAKTIR.