

Hazırlama: 17.12.2023, Son Teslim: 08.01.2024 23:59

## BLM1031 YAPISAL PROGRAMLAMA

### Dönem Projesi

\*\*\*\*\*

## KARŞIT MADDE LABİRENTİ

Bir labirent  $Lab[N][M]$  matrisi ile ifade edilmektedir. Labirentteki yollarda atomaltı parçacıklar vardır. Kullanıcı belirtilen giriş noktasından oyuna başlar ve çıkış noktasına doğru ilerler. Labirentin yollarından toplanan atomaltı parçacıklar kullanılarak karşıt hidrojen üretilmektedir. Labirentte ilerlerken uyulması gereken kurallar ve labirentte bulunan parçacıklar aşağıda sıralanmıştır:

1. Labirentteki yollarda aşağıdaki parçacıklar bulunmaktadır.  
P+ : proton  
e- : elektron  
P- : karşıt proton  
e+ : karşıt elektron
2. Karşıt hidrojen üretebilmek için labirentin çıkışında elinizde sadece P- ve e+ parçacıkları bulunmalıdır.
3. Bir parçacıkla o parçacığın zıt işaretli karşıt parçacığı bir araya gelirse birbirini yok eder.
4. Karadelikler K ile gösterilmiştir. Karadeliklerin bulunduğu hücrelerden geçildiği takdirde oyun sonlanır.
5. Belirli bir oyun süresi içinde kullanıcı çıkışa ulaşamazsa oyun sonlanır.
6. Oyun açıldığında kullanıcının hesap oluşturmaları ya da login olmaları sağlayan bir menü ekrana gelmelidir. Hesap oluşturma adımında kullanıcının isim, soy isim, kullanıcı adı ve şifre bilgileri alınacaktır. Aynı kullanıcı ismine sahip birden fazla kullanıcı olamaz.
7. Login adımında kullanıcının kullanıcı adı ve şifresi ile sisteme girmesi gerekmektedir.
8. Kullanıcıyı oyuna ait en yüksek beş skor ve bu skorlara sahip kullanıcıların görüntülenmesini, oyunun nasıl oynanacağı hakkındaki bilgilerin görüntülenmesini ya da oyuna başlanmasını sağlayan bir menü karşılamalıdır.
9. Oyun, kullanıcı kontrollü ve auto play modunda oynanabilmelidir.
10. Kullanıcı hazır haritalardan birini seçerek oynayabilmeli ya da istediği bir haritayı yükleyebilmelidir. Haritalar text dosya formatında tutulmalıdır.

Kullanıcı kontrolü modunda kullanıcının oyunu kendi istediği dört yöne giderek oynaması sağlanmalıdır. Labirentte gezinmek için ok tuşları kullanılacaktır. Karadelikler üzerine gelirse oyun sonlanır. Kullanıcı oyunda x ile gösterilmelidir. Kullanıcının her hamlesinde ekran temizlenerek kullanıcının güncel lokasyonu x ile işaretlenmelidir. Duvarlara ve matris dışına hamle yapıldığında kullanıcının yeri değişmez. Kullanıcıya istediği zaman esc tuşuna basarak oyundan çıkma hakkı tanınmalıdır. Her hamle ile elde edilmiş atom altı parçacıklar ve labirentten çıkıldığında kaç karşıt madde üretildiği ekrana yazılmalıdır.

Auto play modunda oyun, sizin belirlediğiniz bir algoritmayı kullanarak kendi kendine oynanmalıdır. Yapılan her hamle sonunda oyun alanının son hali ve toplanan atom altı parçacıklar kullanıcı kontrolü modunda olduğu gibi yine ekrana yazılmalıdır.

Labirentin elemanları aşağıdaki gibi gösterilmelidir.

P+	: P	e-	: e	P-	: p	e+	: E
Kara Delik	: K	Kullanıcı: X		Çıkış	: C	Yol	: O
Duvar	: 1	Giriş	: G				

Örnek bir oyun için ekran çıktısı aşağıdaki verilmiştir. (Hazırladığınız kodun renklendirme yapması ve matris sınırlarını çizmesi beklenmemektedir.)

0	0	1	0	P+	0	e-	0	P-	0
1	0	P-	0	1	e+	1	0	1	0
1	X	1	0	1	0	1	0	0	K
0	0	P+	0	P-	0	0	0	1	0
0	1	1	0	K	C	1	0	1	0
0	0	0	0	1	0	P-	0	0	0

Toplanan atom altı parçacıklar:

0	0	1	0	P+	0	e-	0	P-	0
1	X	P-	0	1	e+	1	0	1	0
1	G	1	0	1	0	1	0	0	K
0	0	P+	0	P-	0	0	0	1	0
0	1	1	0	K	C	1	0	1	0
0	0	0	0	1	0	P-	0	0	0

Toplanan atom altı parçacıklar:

0	0	1	0	P+	0	e-	0	P-	0
1	0	X	0	1	e+	1	0	1	0
1	G	1	0	1	0	1	0	0	K
0	0	P+	0	P-	0	0	0	1	0
0	1	1	0	K	C	1	0	1	0
0	0	0	0	1	0	P-	0	0	0

Toplanan atom altı parçacıklar:

P-

0	0	1	0	P+	0	e-	0	P-	0
1	0	0	X	1	e+	1	0	1	0
1	G	1	0	1	0	1	0	0	K
0	0	P+	0	P-	0	0	0	1	0
0	1	1	0	K	C	1	0	1	0
0	0	0	0	1	0	P-	0	0	0

Toplanan atom altı parçacıklar:

P-

0	0	1	X	P+	0	e-	0	P-	0
1	0	0	0	1	e+	1	0	1	0
1	G	1	0	1	0	1	0	0	K
0	0	P+	0	P-	0	0	0	1	0
0	1	1	0	K	C	1	0	1	0
0	0	0	0	1	0	P-	0	0	0

Toplanan atom altı parçacıklar:

P-

0	0	1	0	X	0	e-	0	P-	0
1	0	0	0	1	e+	1	0	1	0
1	G	1	0	1	0	1	0	0	K
0	0	P+	0	P-	0	0	0	1	0
0	1	1	0	K	C	1	0	1	0
0	0	0	0	1	0	P-	0	0	0

Toplanan atom altı parçacıklar:

P- P+

0	0	1	0	0	X	e-	0	P-	0
1	0	0	0	1	e+	1	0	1	0
1	G	1	0	1	0	1	0	0	K
0	0	P+	0	P-	0	0	0	1	0
0	1	1	0	K	C	1	0	1	0
0	0	0	0	1	0	P-	0	0	0

Toplanan atom altı parçacıklar:

P- P+

0	0	1	0	0	0	e-	0	P-	0
1	0	0	0	1	X	1	0	1	0
1	G	1	0	1	0	1	0	0	K
0	0	P+	0	P-	0	0	0	1	0
0	1	1	0	K	C	1	0	1	0
0	0	0	0	1	0	P-	0	0	0

Toplanan atom altı parçacıklar:

P- P+ e+

0	0	1	0	0	0	e-	0	P-	0
1	0	0	0	1	0	1	0	1	0
1	G	1	0	1	X	1	0	0	K
0	0	P+	0	P-	0	0	0	1	0
0	1	1	0	K	C	1	0	1	0
0	0	0	0	1	0	P-	0	0	0

Toplanan atom altı parçacıklar:

P- P+ e+

0	0	1	0	0	0	e-	0	P-	0
1	0	0	0	1	0	1	0	1	0
1	G	1	0	1	0	1	0	0	K
0	0	P+	0	P-	X	0	0	1	0
0	1	1	0	K	C	1	0	1	0
0	0	0	0	1	0	P-	0	0	0

Toplanan atom altı parçacıklar:

P- P+ e+

0	0	1	0	0	0	e-	0	P-	0
1	0	0	0	1	0	1	0	1	0
1	G	1	0	1	0	1	0	0	K
0	0	P+	0	P-	0	X	0	1	0
0	1	1	0	K	C	1	0	1	0
0	0	0	0	1	0	P-	0	0	0

Toplanan atom altı parçacıklar:

P- P+ e+

0	0	1	0	0	0	e-	0	P-	0
1	0	0	0	1	0	1	0	1	0
1	G	1	0	1	0	1	0	0	K
0	0	P+	0	P-	0	0	X	1	0
0	1	1	0	K	C	1	0	1	0
0	0	0	0	1	0	P-	0	0	0

Toplanan atom altı parçacıklar:

P- P+ e+

0	0	1	0	0	0	e-	0	P-	0
1	0	0	0	1	0	1	0	1	0
1	G	1	0	1	0	1	0	0	K
0	0	P+	0	P-	0	0	0	1	0
0	1	1	0	K	S	1	X	1	0
0	0	0	0	1	0	P-	0	0	0

Toplanan atom altı parçacıklar:

P- P+ e+

0	0	1	0	0	0	e-	0	P-	0
1	0	0	0	1	0	1	0	1	0
1	G	1	0	1	0	1	0	0	K
0	0	P+	0	P-	0	0	0	1	0
0	1	1	0	K	C	1	0	1	0
0	0	0	0	1	0	P-	X	0	0

Toplanan atom altı parçacıklar:

P- P+ e+

0	0	1	0	0	0	e-	0	P-	0
1	0	0	0	1	0	1	0	1	0
1	G	1	0	1	0	1	0	0	K
0	0	P+	0	P-	0	0	0	1	0
0	1	1	0	K	C	1	0	1	0
0	0	0	0	1	0	X	0	0	0

Toplanan atom altı parçacıklar:

P- P+ e+ P-

0	0	1	0	0	0	e-	0	P-	0
1	0	0	0	1	0	1	0	1	0
1	G	1	0	1	0	1	0	0	K
0	0	P+	0	P-	0	0	0	1	0
0	1	1	0	K	C	1	0	1	0
0	0	0	0	1	X	0	0	0	0

Toplanan atom altı parçacıklar:

P- P+ e+ P-

0	0	1	0	0	0	e-	0	P-	0
1	0	0	0	1	0	1	0	1	0
1	G	1	0	1	0	1	0	0	K
0	0	P+	0	P-	0	0	0	1	0
0	1	1	0	K	X	1	0	1	0
0	0	0	0	1	0	0	0	0	0

Çıkışa ulaşıldı

1 adet karşıt madde üretildi

Kullanıcın girdiği esc ve ok tuşlarının kodlarını ekrana bastıran ve her seferinde ekranı temizleyen örnek bir kod parçası aşağıda verilmiştir.

```
#include <stdlib.h>
#include <stdio.h>
#include <unistd.h>
int main()
{
    int i,j,k;
    char a;
    for (i=10;i>0;i--)
    {
        printf("Geri Sayim:%d\n",i);
        a = getch();
        if (a==27)
        {
            printf("Cikis\n");
            break;
        }
        else
        {
            a = getch();
            printf("Bastiginiz ok tusu:%d\n",a);
        }
        sleep(1); system("cls");
    }
    printf("BOOOOM!"); return 0;
}
```

Proje dâhilindeki tüm kodlar fonksiyonlar halinde hazırlanmalıdır. Oyun haritası için gerekli alanlar dynamic memory allocation ile açılmalıdır. Kullanıcı bilgileri structure yapısında tutulmalı ve binary formatta kaydedilmelidir. Hazırlanan kaynak kod işlem blokları ve değişkenler için yazılmış açıklama satırları içermelidir. Proje kontrolü final haftası daha sonra ilan edilecek gün, yer, saat ve şekilde yapılacaktır. Olası değişiklikler için online.yildiz ve OBS sayfalarınızı takipte kalınız.