

Panel De Pon

BLM1011 Dönem Projesi

Alperen Ateş

25011018

Video Linki: https://youtu.be/s_MKI5f6wp4

Panel De Pon Nedir?

5 çeşit taş ve $N \times M$ 'lik bir tahtadan oluşan oyundur. Oyun alttan $N/2$ satır dolu şekilde başlar. Oyuncu yan yana bulunan farklı tür taşların yerini değiştirebilir, aynı tür taştan yan yana veya alt alta en az 3 tane olmak koşuluyla taşları patlatabilir. Taşlar patlatıldığında boş kalan yerlere üst satırlardaki taşlar düşer. Yer değiştirme sonucu tüm taşlar bir üst satıra kayar ve en alta rastgele karakterlerden oluşan yeni bir satır eklenir. Oyun en üst satırda herhangi bir taş bulunduğuunda veya yapılabilecek hamle kalmadığında biter. Oyuncunun amacı en fazla miktarda taş patlatmaktır.

Oyunun Özeti Algoritması:

- 1- Oyuncudan matrisin boyutunu (N, M) ve oyun modunu al.
- 2- Oyun modu "normal" mod ise alttan $n/2$ satırı rastgele oluştur. Oyun modu "kontrol" mod ise alttan $n/2$ satırı oyuncudan iste, matrisi oluştur.
- 3- Oyun tahtası, yer değiştirme sayısı ve patlama sayısı ekrana yazılır.
- 4- Oyuncudan yapmak istediği işlem alınır: yer değiştirme, patlatma veya oyunu terk etme.
 - Yer değiştirme:
 - 5- Oyuncudan hangi iki taşın yerini değiştirmek istediği bilgisi alınır.
 - 6- Yerleri girilen iki koordinat yer değiştirebiliyorsa yer değiştirilir.
 - 7- Oyuncuya yer değiştirme işleminin yapıldığı bilgisi verilir. Kaydırma ve en aşağı yeni bir rastgele satır ekleme işlemi için oyuncu beklenir.
 - 8- Oyun tahtası aşağıdan yukarıya kaydırılır ve en aşağı rastgele yeni satır eklenir. Bu işlem yapılırken aynı zamanda herhangi bir sütunda bir taşın en üst sütuna ulaşıp ulaşmadığı kontrol edilir.
 - 9- Eğer en üstte bir taş varsa oyun sonlandırılır.
 - Patlatma:
 - 5- Oyuncudan patlamanın nerede olacağı bilgisi alınır.
 - 6- Patlamanın yönü belirlenir.

7- Patlama gerçekleşebiliyorsa gerçekleşir ve yer çekimi uygulanması için oyuncu beklenir.

8- Yer çekimi uygulanır.

9- Oyunun kilitlenip kilitlenmediği kontrol edilir. Kilitlendiye oyuncu bilgilendirilip oyun sonlandırılır.

```
else if(gameOperation == GAME_OPERATION_MOVE) /* İki karakterin yerini değiştirmek istediğiinde */
{
    MovementCoords movementCoords = get_movement_coordinates(&gameBoard, gameStatus);
    if(movementCoords.c1.i != -1) /* işlem iptal edilmedi */
    {
        do_movement(&gameBoard, movementCoords);

        gameStatus.movementCount++;

        print_status(&gameBoard, gameStatus);
        printf("Yer değiştirildi. Yeni satır ekleniyor\n");
        pause();

        ended = do_append_new_random_row(&gameBoard);

        if(ended == 1) /* en üst satırda 0 olmayan eleman var */
        {
            print_status(&gameBoard, gameStatus);
            printf("Oyun en üst satırda eleman bulunduğu için sonlandırılmıştır.\n");
        }
    }
}
```

Oyuncu yer değiştirmek istediğiinde çalışan kod

```
else /* operation == GAME_OPERATION_EXPLODE, ancak başka bir şey olamayacağı için direkt else dedik */
{
    Coord coord = get_explosion_coordinate(&gameBoard, gameStatus);
    if(coord.i != -1) /* kullanıcı işlemi iptal etmediye */
    {
        ExplosionInfo explosionInfo = {EXPLOSION_DIRECTION_NOT_VALID, {0, 0}, 0}; /* ilk değeri bilmiyoruz */
        explosionInfo.coord = coord;
        do_explosion(&gameBoard, &explosionInfo);
        if(explosionInfo.range > 0) /* patlama olmadıysa yer çekimi uygulamaya ve kilitlenme kontrolü yapmaya gerek yok */
        {
            gameStatus.explosionCount += explosionInfo.range;

            print_status(&gameBoard, gameStatus);
            printf("Patlama gerçekleşti. Yer çekimi uygulanıyor\n");
            pause();

            do_gravity(&gameBoard, &explosionInfo);

            ended = !do_check_possible_mov_and_exp(&gameBoard); /* hareket mümkün değilse biter */

            /* sadece patlama durumunda oyun kilitlenebilir */
            if(ended == 1)
            {
                print_status(&gameBoard, gameStatus);
                printf("Oyun kilitlendi!\n");
            }
        }
    }
    else /* patlama olmadığını kullanıcıya bildir */
    {
        printf("Herhangi bir patlama olamadı!\n");
        pause();
    }
}
```

Oyuncu patlama istediğiinde çalışacak kod

Kullanılan Standart Kütüphaneler, Fonksiyonları ve Makroları:

- 1) stdio.h
 - a) printf, puts: Output amacıyla kullanılan fonksiyonlar
 - b) scanf, getchar: Input amacıyla kullanılan fonksiyonlar
- 2) stdlib.h
 - a) srand, rand: rastgele sayılar üretmek için kullanılan fonksiyonlar
 - b) system: Terminal ekranın temizlemeyi sağlayan komutu terminale gönderen fonksiyon.
 - c) abs: mutlak değer fonksiyonu, iki taşın yan yana olup olmadığı kontrol edilmek için kullanılmıştır.
- 3) time.h
 - a) time: rastgele sayı üreten fonksiyonun başlangıç değerini belirlemek için zaman kullandık. Duyarlılığı 1 saniye kadardır.
- 4) assert.h
 - a) assert: oluşturup kullandığım fonksiyonlarda pointer hatası oluşmaması için kullanılan makro (son üründe NDEBUG ile etkisiz hale getirildi).

```
#include <stdio.h> /* printf, puts, scanf, getchar */  
#include <stdlib.h> /* rand, srand, system, abs */  
#include <time.h> /* time */  
#include <cassert.h> /* assert */
```

Dahil edilen kütüphaneler, ve kullanılan fonksiyonları listeleyen yorumlar

Kullanılan Standartlar, Yazı Stilleri ve Fonksiyon İsimlendirmeleri

- Dil standardı olarak ANSI C (C89) kullanılmıştır.
- Tür isimlendirmesi olarak PascalCase kullanılmıştır
- Fonksiyon isimlendirmesi olarak snake_case kullanılmıştır.
- Değişken isimlendirmesi olarak camelCase kullanılmıştır.
- Sabit ve makro isimlendirmesi olarak UPPER_CASE kullanılmıştır.
- clear ön ekine sahip fonksiyonlar bir veriyi temizler.
- print ön ekine sahip fonksiyonlar bir veriyi kullanıcıya gösterir.
- get ön ekine sahip fonksiyonlar bir veriyi kullanıcıdan alır.
- do ön ekine sahip fonksiyonlar oyunun hesaplama içeren işlevlerini yerine getirir.

```
#define MAX_GAME_MATRIX_SIZE    20 /* En büyük matris boyutu */
#define MIN_GAME_MATRIX_SIZE     3  /* En küçük matris boyutu */

#define MAX_GAME_MATRIX_POSITION 20 /* Yer değiştirme veya patlama için girilebilecek en büyük koordinat */
#define MIN_GAME_MATRIX_POSITION 1  /* Yer değiştirme veya patlama için girilebilecek en küçük koordinat */

#define GAME_CHARACTERS_LENGTH 5 /* oyundaki toplam karakter çeşiti sayısı */

#define MIN_EXPLOSION_RANGE 3 /* en az bu kadar taş aynı tipte olmalı ki patlama olabilsin */

/* Oyun ön bilgisindeki n/2 satırının indisini bulan makro */
#define MIDDLE_ROW(n) ((n)-(n)/2)
```

Sabitler ve makrolar

```
/* Oyun modlarının alabileceği değerler */
typedef enum GameMode
{
    GAME_MODE_UNKNOWN = 0, /* henüz bir değer almadığını göstermek için */
    GAME_MODE_NORMAL = 1,
    GAME_MODE_CONTROL = 2
}GameMode;

/* Oyun döngüsünde kullanıcının yapabileceği işlemler */
typedef enum GameOperation
{
    GAME_OPERATION_EXIT = -1,
    GAME_OPERATION_UNKNOWN = 0, /* henüz bir değer almadığını göstermek için */
    GAME_OPERATION_MOVE = 1,
    GAME_OPERATION_EXPLODE = 2
}GameOperation;

typedef enum ExplosionDirection
{
    EXPLOSION_DIRECTION_NOT_VALID = 0, /* patlama olamaz demek */
    EXPLOSION_DIRECTION_RIGHT = 1,
    EXPLOSION_DIRECTION_DOWN = 2
}ExplosionDirection;
```

Tipli sabitler

```
/* Stdin'deki çöp değerleri temizler */
> void clear_stdin() ...

/* Terminal ekranını temizler */
> void clear_screen() ...

/* oyun kurallarına göre en başta boş olması gereken yerlere '\0' karakterlerini doldurur
> void clear_matrix(GameBoard *gameBoard) ...
```

clear ön ekine sahip fonksiyonlar

```
/* Oyun ilk açıldığında ekrana yazılacak yazıları barındıran fonksiyon */
> void print_splash_screen() ...

/* Oyun tahtasındaki matrisi boyuta uygun bir biçimde ekrana yazdırır */
> void print_matrix(const GameBoard *gameBoard) ...

/* Oyunun o anki durumunu yazan fonksiyon, parametre olarak verilen değerlere
> void print_status(const GameBoard *gameBoard, GameStatus gameStatus) ...
```

print ön ekine sahip fonksiyonlar

```
> MatrixSize get_matrix_size(void) ...

/* Kullanıcıdan oyun modu istenir */
> GameMode get_game_mode(void) ...

/* Kullanıcıdan yer değişikliği mi patlama mı istediği sorulur */
> GameOperation get_game_operation(void) ...

/* Eğer işlem kullanıcı tarafından iptal edilmek istenirse (rakam olmayan bir karaktere basılırsa)
> MovementCoords get_movement_coordinates(const GameBoard *gameBoard, GameStatus gameStatus) ...

/* Eğer işlem kullanıcı tarafından iptal edilmek istenirse (rakam olmayan bir karaktere basılırsa)
> Coord get_explosion_coordinate(const GameBoard *gameBoard, GameStatus gameStatus) ...
```

get ön ekine sahip fonksiyonlar

```
/* Oyun moduna göre gameMatrix'i kullanıcıdan alır veya rastgele oluşturur */
> void do_initialize_game_matrix(GameBoard *gameBoard, GameMode gameMode) ...

/* ExplosionInfo'ya göre yer çekimi etkisini uygular */
> void do_gravity(GameBoard *gameBoard, ExplosionInfo *explosionInfo) ...

/* Verilen konuma göre patlama olup olamayacağını, olursa ne yöne olacağını ayarlar. ve patlama olacak ye
> void do_explosion(GameBoard *gameBoard, ExplosionInfo *explosionInfo) ...

/* geri dönüş değeri olarak oyunun bitip bitmediğini döner (0 bitmedi, 1 bitti) */
> int do_append_new_random_row(GameBoard *gameBoard) ...

/* İki taşın yerini değiştiren fonksiyon */
> void do_movement(GameBoard *gameBoard, MovementCoords movementCoords) ...

/* Herhangi bir patlama veya yer değişikliği mümkün mü diye kontrol eder. (0 mümkün değil, 1 mümkün) */
> int do_check_possible_mov_and_exp(GameBoard *gameBoard) ...
```

do ön ekine sahip fonksiyonlar

```

int main()
{
    int ended = 0;
    GameBoard gameBoard = {0};
    GameStatus gameStatus = {GAME_MODE_UNKNOWN, 0, 0};
    GameOperation gameOperation = GAME_OPERATION_UNKNOWN;
    MatrixSize matrixSize = {0}; /* get_matrix_size fonksiyonu

```

camelCase ile tanımlanan değişkenler

```

/* Oyun ilk açıldığında ekrana yazılacak yazıları barındıran fonksiyon */
void print_splash_screen()
{
    /* stringleri parçalara ayırdık çünkü bu kadar uzun bir string c89 (ansi c) tarafından desteklenmez (null hariç en fazla 509 karakter) */
    puts(
        "Panel De Pon oyununa hoş geldiniz!\n"
        "Bu oyunun amacı karakterler en üst satırda değişmeden en fazla sayıda element patlatmaya çalışmaktadır\n"
        "Oyun karakterleri: [*/+XO]\n"
        "\n"
        "Oyun kuralları: \n"
        "  1- İki elementi yer değiştirmek istediğinizde sadece komşu iki elementlerin yerlerini değiştirebilirsiniz\n"
        "  2- Yerleri değiştirecek elemanlar birbirinden farklı olmalıdır\n"
        "  3- İki elementin yerleri değiştirildiğinde tüm satırlar birer yukarı kayar ve en alt satırda rastgele elementler eklenir"
    );
    puts(
        "  4- En üst satırda herhangi bir element ulaşılığında oyun direkt olarak sona erer\n"
        "  5- Bir element grubunu patlatabilmek için en az 3 tane yan yana veya alt alta aynı elementten bulunmalıdır\n"
        "  6- Eğer hem dikey hem yatay patlama olabiliyorsa oyun en fazla element patlayabilecek olanı seçecektir\n"
        "  7- Eğer dikey ve yatay patlayabilecek element sayısı eşitse oyun yatay olamı secer"
    );
    puts(
        "  8- Bir element grubunu patlatmaya karar verdığınızda bu grubun en üst veya en sol elementinin koordinatını girmeniz gerekmektedir\n"
        "  9- Yer değiştirme veya patlama işlemi seçildikten sonra koordinat yerine rakam olmayan bir karakter girmek işlemi iptal etmeyi sağlar\n"
        "\n"
        "Oyun Modları: \n"
        "  1- Normal Mod:\n"
        "    Bu modda oyun matrisinin alttan n/2 kadar satırı rastgele elementlerle doldurulur.\n"
        "    Her hamleden sonra oyun ekranı temizlenir"
    );
    puts(
        "  2- Kontrol Mod:\n"
        "    Bu modda oyun matrisinin alttan n/2 kadar satırı ve M kadar sütununda bulunan elementler kullanıcının istenir"
    );
}

```

print_splash_screen fonksiyonu ve ANSI C 509 karakter sınırlandırması

Kaynak: [ANSI C 2.2.4.1 Translation Limits](#)

Oyun İçin Oluşturulan Veri Tipleri:

- 1) GameMatrix: Oyun matrisinin türüdür, 20x20'lik char tipinden bir matris.
- 2) GameBoard: Fonksiyon parametrelerini sadeleştirmek için kullanıldı.
 - a. GameMatrix: 20x20'lik oyun matrisinin kendisi
 - b. n: Oyun matrisinin mantıksal satır sayısı
 - c. m: Oyun matrisinin mantıksal sütun sayısı
- 3) Coord: Coordinate'nin kısaltılmıştır, oyun tahtasındaki bir noktayı göstermek için kullanılır
 - a. i: satır bilgisi
 - b. j: sütun bilgisi
- 4) MatrixSize: Kullanıcıdan matrisin boyutunu tek seferde alabilmek için oluşturulan tür, matrisin boyut bilgisini tutar.
 - a. n: Matrisin satır sayısı
 - b. m: Matrisin sütun sayısı
- 5) ExplosionInfo: Patlama bilgisini tutan tür, fonksiyonlara bilgileri tek seferde gönderebilme için kullanıldı
 - a. explosionDirection: Patlamanın yönünü tutar.
 - b. coord: Patlamanın konumunu tutar.
 - c. range: patlamanın ne kadar uzunlukta etkili olduğunu tutar.
- 6) MovementCoords: Hareket ettirilecek iki taşın konum bilgisini tutan yapı.
 - a. c1: 1. koordinat
 - b. c2: 2. koordinat
- 7) GameStatus: Oyun durumunu tutan yapı.
 - a. gameMode: Oyuncunun normal mod mu kontrol mod mu seçtiği bilgisini tutar.
 - b. movementCount: Yapılan yer değişikliği sayısını tutar.
 - c. explosionCount: Patlatılan toplam taş sayısını tutar.

```

/* Oyun matrisini ve matrisin boyutunu içinde tutan yapı */
typedef struct GameBoard
{
    GameMatrix matrix;
    int n; /* Row */
    int m; /* Column */
}GameBoard;

/* Oyun tahtasındaki bir noktayı gösteren koordinatı temsil eden yapı */
typedef struct Coord
{
    int i; /* Row */
    int j; /* Column */
}Coord;

typedef struct MatrixSize
{
    int n; /* Row */
    int m; /* Column */
}MatrixSize;

typedef struct ExplosionInfo
{
    ExplosionDirection explosionDirection;
    Coord coord;
    int range;
}ExplosionInfo;

/* Hareket hamlesi için gerekli olan iki koordinatı saklayan yapı */
typedef struct MovementCoords
{
    Coord c1;
    Coord c2;
}MovementCoords;

/* Oyun durumunu tek paket içinde tutmamızı sağlayan yapı */
typedef struct GameStatus
{
    GameMode gameMode;
    int movementCount, explosionCount;
}GameStatus;

```

Kodun okunabilirliğini arttırmak için kullanılan yapı türleri

Oyun İçin Oluşturulan Fonksiyonlar

1) clear ön ekine sahip olanlar:

Fonksiyon	Parametreler	Dönüş Türü	Açıklama (özet)	Açıklama (detay)
clear_stdin	Yok	Yok	Standart girdi tamponunu temizler.	scanf veya getchar ile bir girdi alındığı zaman bu fonksiyonların işlemediği ama kullanıcının klavyeden girmiş olduğu karakterler girdi tamponu olarak kalır. Eğer bu fonksiyon ile temizlenmezler ise bir sonraki scanf veya getchar fonksiyonun sabote edebilir.
clear_screen	Yok	Yok	Terminal ekranını temizler.	Kaynak kodun derlendiği platforma göre uygun komut satırı çağrıları ile terminal ekranı temizlenir. Eğer kodda tanımlanmamış bir platform ile karşılaşılırsa 1000 tane '\n' karakteri ile terminal ekranı "temizlenmiş" gibi yapılır.
clear_matrix	GameBoard*	Yok	Matrisi temizler.	Parametre olarak gönderilen oyun tahtasındaki matrisi, oyun tahtasının satır ve sütun sayısına göre oyunun başında boş ('\0') olması gereken yerleri temizler.

2) print ön ekine sahip olanlar:

Fonksiyon	Parametreler	Dönüş Türü	Açıklama (özet)	Açıklama (detay)
print_splash_screen	Yok	Yok	Oyunun giriş ekranını yazar.	Program çalıştırıldığında kullanıcıya oyunun kurallarını ve oyunun modlarını bildirir. Bu işlem karakter dizileri ile yapılır ve C dilinin standardına göre bir karakter dizisinin uzunluğu 509 karakter olabilir (null hariç). Bu yüzden bu fonksiyonda kullanılan karakter dizileri parça parça yazılmıştır.
print_matrix	GameBoard*	Yok	Matrisi okunabilir formatta yazar.	Matrisin sütun numaralarının onlar basamağını başlangıç hizalarına yazar. Bir alt satırına sütun numaralarının birler basamağındaki rakamlar yazılır. Sütun numaraları yazıldıktan sonra her satır arasına '-' karakteri gelecek şekilde önce satır numarası daha sonra satırındaki karakterler aralarında ' ' karakteri bulunacak şekilde yazılır. Sütun numaralarının tek satırda değil de ayrı satırlarda yazılmasının sebebi sütun numarası 10'dan büyük olduğu durumda sütun ile hizalanamadığı içindir.

print_status	GameBoard*, GameStatus	Yok	Her turda ekraná yazılması gerekenleri yazar.	Parametre olarak verilen GameStatus yapısındaki verilere göre oyun modu "normal" mod ise clear_screen ile ekranı temizler. Ardından yapılan toplam yer değişikliği ile toplam patlatılan eleman sayısını yazdırır. En sonunda ise print_matrix aracılığıyla ekraná oyun matrisini yazdırır.
--------------	------------------------	-----	---	---

3) get ön ekine sahip olanlar:

Fonksiyon	Parametreler	Dönüş Türü	Açıklama (özet)	Açıklama (detay)
get_matrix_size	Yok	MatrixSize	Oyuncudan matrisin boyutunu istenir.	Kullanıcıdan oyun matrisinin boyutunu ister. Girilen boyut "satır sütun" şeklinde olmalıdır. Girilen boyut değerleri geçerli bir değer ve aralıktaki bulunmadığı sürece kullanıcıdan boyut bilgisini istemeye devam eder. Geçerli bir koordinat girildiğinde MatrixSize yapısı ile döndürülür.
get_game_mode	Yok	GameMode	Oyuncudan oyunun modu istenir.	Kullanıcıdan oyun modunu ister. İki çeşit oyun modu vardır: 1- Normal mod (arayüzde "oyun modu" olarak geçer) ve Kontrol mod. Geçerli bir mod girilmediği sürece kullanıcıdan mod bilgisini istemeye devam eder. Kullanıcının girdiği geçerli oyun modunu döndürür.
get_game_operation	Yok	GameOperation	Oyuncudan yer değişikliği mi patlama mı yapmak istediği bilgisi istenir.	Kullanıcıdan yapılacak işlemi ister. Yapılabilecek işlemler şunlardır: 1) Yer değişikliği, 2) Patlama, -1) Oyundan ayrılma. Kullanıcı bu işlemlerden birini seçmediği sürece işlem bilgisi istemeye devam eder. Kullanıcının girdiği geçerli işlem döndürür.
get_movement_coordinates	GameBoard*, GameStatus	MovementCoordinates	Oyuncudan yer değiştirmek istediği iki koordinat alınır. Bu iki koordinatın oyun kuralları içerisinde yer değiştirebilip değiştiremeyeceğini kontrol eder. Eğer yerler değiştirilemezse kullanıcı bilgilendirilip koordinatlar bir daha istenir. Kullanıcı yer değiştirme işlemini iptal etmek isterse rakam olmayan bir girdi girmesi yeterlidir. İşlem iptal edildiğinde -1 koordinatlarını döndürür, bu değer main tarafından işlemin iptal edildiği şeklinde yorumlanır. Kurallara uygun girdilerde ise girilen koordinatları döndürür. Doğru girdi formatı şu şekildedir: satır1,sütun1 satır2,sütun2	Kullanıcıdan yerlerini değiştirmek istediği iki koordinat alınır. Bu iki koordinatın oyun kuralları içerisinde yer değiştirebilip değiştiremeyeceğini kontrol eder. Eğer yerler değiştirilemezse kullanıcı bilgilendirilip koordinatlar bir daha istenir. Kullanıcı yer değiştirme işlemini iptal etmek isterse rakam olmayan bir girdi girmesi yeterlidir. İşlem iptal edildiğinde -1 koordinatlarını döndürür, bu değer main tarafından işlemin iptal edildiği şeklinde yorumlanır. Kurallara uygun girdilerde ise girilen koordinatları döndürür. Doğru girdi formatı şu şekildedir: satır1,sütun1 satır2,sütun2

get_explosion_coordinate	GameBoard*, GameStatus	Coord	Oyuncudan patlama yapmak istediği koordinat alınır.	Kullanıcıdan bir koordinat bilgisi ister. Bu koordinatın oyun sınırları içerisinde olup olmadığını kontrol eder. Eğer sınır içerisinde ise bu koordinatın boş olup olmadığını kontrol eder. Kullanıcı işlemi iptal etmek isterse rakam olmayan bir girdi girmesi yeterlidir. İşlem iptal edildiğinde -1 koordinatı döndürülür, bu değer main tarafından işlemin iptal edildiği şeklinde yorumlanır. Girdi kurallara uygunsa girilen koordinat döndürülür. Bu fonksiyon patlamanın gerçekleşip gerçekleşmeyeceğini kontrol etmez, sadece sınırları ve o koordinatta bulunan karakteri kontrol eder. Doğru girdi formatı: satır,sütun
--------------------------	------------------------	-------	---	---

4) do ön ekine sahip olanlar:

Fonksiyon	Parametreler	Dönüş Türü	Açıklama (özet)	Açıklama (detay)
do_initialize_game_matrix	GameBoard*, GameMode	Yok	Oyun moduna göre oyun tahtasına ilk değerleri atar.	Önce clear_matrix ile matrisi temizler. Sonra önceden belirlenmiş oyun moduna göre matrisin alttan n/2 satırını doldurur. Eğer normal modda ise matrisi önceden tanımlanmış oyun karakterlerinin bulunduğu diziden rastgele bir eleman seçerek oluşturur. Eğer kontrol modda ise matrisin doldurulacak kısımları kullanıcidan istenir. Kullanıcı hatalı bir karakter girerse doğru bir karakter girene kadar aynı koordinatın değeri istenmeye devam eder.
do_gravity	GameBoard*, ExplosionInfo*	Yok	Patlama durumuna göre yer çekimi uygular.	Parametre olarak aldığı ExplosionInfo'ya göre yer çekimi uygular. Eğer patlama sağa doğru olmuşsa patlama olan sütunlarda patlama olan satırın üstünü birer satır aşağı kaydırır. Eğer patlama aşağı doğru oldusaya patlama olan satırın üstünde bulunan elemanları patlama uzunluğu kadar aşağı öteler. Aşağı itilen karakterlerin olduğu koordinatlara ise '\0' karakteri yerleştirir.

do_explosion	GameBoard*, ExplosionInfo*	Yok	Verilen koordinata göre patlama yapar.	Parametre olarak aldığı ExplosionInfo yapısındaki koordinata bakarak önce sağa doğru ve aşağı doğru patlama olması durumunda kaç eleman patlayacağını hesaplar. Sonra bu mesafelerden büyük olanının 3'ten büyük olması durumunda ExplosionInfo yapısını bu mesafeye ve yönे göre ayarlar. En sonunda da patlama olan yerlere '\0' değerini atar. 1) Eğer patlama olmazsa ExplosionInfo'da patlama yönü ve büyülüğu 0 olur. 2) Eğer sağa ve aşağı patlama aynı büyüklükte olacaksa sağa doğru patlama tercih eder, Bunun sebebi yatay patlamanın dikey patlamaya göre daha fazla sütunu en üst satırдан uzaklaştırmasıdır. 3) Patlama olan yerlere '\0' karakterinin yazılması sadece görsel amaçlıdır, kullanıcının patlama olduktan sonraki yer çekimi uygulanmadan önceki anda patlama durumunu görebilmesini sağlar.
do_append_new_random_row	GameBoard*	int	Oyun tahtasının tüm satırlarını bir üste kaydırır ve en alta rastgele bir satır ekler.	Verilen matrisi sütun sütun, en alt satırda başlayacak şekilde bir üst satır örteker. Öteleme işlemini sütunda '\0' karakteri görene kadar yapar, böylece az eleman bulunan sütunlarda gereksiz işlemlerden kaçınır. Sütun yukarı ötelendikten sonra sütunun en alt satırına rastgele bir oyun karakteri eklenir. Eğer sütunun en üst satırında '\0' dışında bir eleman varsa oyunun bittiği bilgisini kaydeder. Bu bilgiyi tüm kaydırma işlemleri bittikten sonra çağrılan fonksiyona döndürür.
do_movement	GameBoard*, MovementCoords	Yok	Verilen taşların yerini değiştirir.	MovementCoords ile verilen iki koordinattaki elemanların yerini değiştirir.
do_check_possible_mov_and_exp	GameBoard*	int	Oyunun kilitlenip kilitlenmediğini kontrol eder.	Verilen oyun matrisine göre kurallar dahilinde yapılabilecek hamle olup olmadığını kontrol eder. Bunu yaparken önce yan yana olan elemanlara bakarak bu elemanların yer değiştirebilir olup olmadığını kontrol eder. Eğer herhangi bir karaktere yer değiştirme işlemi uygulanamıyorsa herhangi üç elemanın patlama yapıp yapamadığını kontrol eder (önce yatay sonra dikey). Bu kontroller sonucu hamle yapılabılıp yapılamama bilgisini çağırılan fonksiyona döner. 1) Önce yer değiştirme kontrolü yapıldı çünkü yer değiştirme işlemi patlama işleminden daha olası bir ihtimaldir. 2) Yer değiştirme ve patlama kontrolü yapılırken önce en alt satırda başlandı, çünkü alt satırlarda '\0' harici bir karakter bulunma ihtimali üst satırlara göre daha fazladır.

main Fonksiyonunun Detaylı Açıklaması

```
int ended = 0;
GameBoard gameBoard = {0};
GameStatus gameStatus = {GAME_MODE_UNKNOWN, 0, 0};
GameOperation gameOperation = GAME_OPERATION_UNKNOWN;
MatrixSize matrixSize = {0}; /* get_matrix_size fonksiyonu
```

Kullanılan değişkenlerin tanımlanması (ANSI C standardına göre kapsam başında olmalı)

```
    srand(time(NULL)); /* rand() fonksiyonuna rastgele sayılar sağlar */

    clear_screen();
    print_splash_screen();
```

Oyuna hazırlık işlemleri:

srand ile rand fonksiyonunun kullandığı değişkene ilk değer atanmış olur.

clear_screen ile terminalde önceden kalmış yazılar temizlenir.

print_splash_screen ile kullanıcıyı oyun hakkında bilgi veren ekran ile karşılamış olur.

```
/* matrisin boyutunu al ve gameBoard'ın değişkenlerine ata */
matrixSize = get_matrix_size();
gameBoard.n = matrixSize.n;
gameBoard.m = matrixSize.m;

gameStatus.gameMode = get_game_mode();

/* Modlara göre matrisi oluştur veya iste */
do_initialize_game_matrix(&gameBoard, gameStatus.gameMode);
```

Kullanıcıdan matrisin boyutu ve oyun modu alınır. Bu bilgilere göre matris oluşturulur veya istenir.

```
/* Oyunun asıl döngüsü */
do
{
    print_status(&gameBoard, gameStatus);

    gameOperation = get_game_operation();
```

Oyunun o anki durumu gösterilir ve kullanıcıdan yapmak istediği işlem alınır.

```
if(gameOperation == GAME_OPERATION_EXIT) /* Oyundan çıkışmak istendiğinde */
{
    printf("Oyun kullanıcı istediği ile sonlandırılmıştır.\n");
    ended = 1;
}
```

Kullanıcı oyundan ayrılmak istediginde çalışan kod

```
else if(gameOperation == GAME_OPERATION_MOVE) /* İki karakterin yerini değiştirmek istedig
{
    MovementCoords movementCoords = get_movement_coordinates(&gameBoard, gameStatus);
```

Kullanıcı yer değişikliği istediginde ilk olarak koordinatlar alınır.

```
if(movementCoords.c1.i != -1) /* işlem iptal edilmemi */
{
    do_movement(&gameBoard, movementCoords);

    gameStatus.movementCount++;

    print_status(&gameBoard, gameStatus);
    printf("Yer değiştirildi. Yeni satır ekleniyor\n");
    pause();

    ended = do_append_new_random_row(&gameBoard);

    if(ended == 1) /* en üst satırda 0 olmayan eleman var */
    {
        print_status(&gameBoard, gameStatus);
        printf("Oyun en üst satırda eleman bulunduğu için sonlandırılmıştır.\n");
    }
}
```

Kullanıcı eğer işlemi iptal etmediyse önce koordinatların yerleri değiştirilir ardından kullanıcının Enter (Return) tuşuna basması beklenir. Daha sonra matrisin elemanları birer satır yukarı kaydırılıp en alta rastgele bir satır eklenir. Eğer en üst satırda '0' haricinde bir eleman varsa kullanıcıya bilgi verilip oyun sonlandırılır.

```
else if(gameOperation == GAME_OPERATION_EXPLODE) /* patlama yapmak istediginde */
{
    Coord coord = get_explosion_coordinate(&gameBoard, gameStatus);
    if(coord.i != -1) /* kullanıcı işlemi iptal etmediyse */
    {
```

Kullanıcı patlama yapmak istediginde patlama koordinatı alınır. Eğer işlem iptal edildiyse {-1, -1} koordinatı döner.

```

ExplosionInfo explosionInfo = {EXPLOSION_DIRECTION_NOT_VALID, {0, 0}, 0}; /* ilk değeri bilmiyoruz */
explosionInfo.coord = coord;
do_explosion(&gameBoard, &explosionInfo);
if(explosionInfo.range > 0) /* patlama olmadıysa yer çekimi uygulamaya ve kilitlenme kontrolü yapmaya gerek yok */
{
    gameStatus.explosionCount += explosionInfo.range;

    print_status(&gameBoard, gameStatus);
    printf("Patlama gerçekleşti. Yer çekimi uygulanıyor\n");
    pause();

    do_gravity(&gameBoard, &explosionInfo);

    ended = !do_check_possible_mov_and_exp(&gameBoard); /* hareket mümkün değilse biter */

    /* sadece patlama durumunda oyun kilitlenebilir */
    if(ended == 1)
    {
        print_status(&gameBoard, gameStatus);
        printf("Oyun kilitlendi!\n");
    }

}
else /* patlama olmadığını kullanıcıya bildir */
{
    printf("Herhangi bir patlama olamadı!\n");
    pause();
}

```

If bloğunun başında patlama bilgisini tutan ExplosionInfo tanımlanır. Bu yapı geçerli koordinatlar atanıktan sonra patlama işlemini hesaplayan do_explosion fonksiyonuna gönderilir. Eğer patlama olduysa oyun tahtası güncellenir ve kullanıcının Enter (Return) tuşuna basması beklenir. Sonrasında patlama yönü ve doğrultusuna göre do_gravity ile yer çekimi uygulanır. Son olarak do_check_possible_mov_and_exp ile herhangi bir geçerli hamle kalıp kalmadığı kontrol edilir.

```

}
}while(ended == 0);

```

Bu işlemler ended değeri 0 olduğu sürece (yani oyun bitmediği sürece) devam eder.

Oyun Ekran Görüntüleri

1. Örnek

Panel De Pon oyununa hoş geldiniz!
Bu oyunun amacı karakterler en üst satırda değişmeden en fazla sayıda element patlatmaya çalışmaktadır
Oyun karakterleri: [*/+%]

Oyun kuralları:

- 1- İki elementi yer değiştirmek istediğinizde sadece komşu iki elementlerin yerlerini değiştirebilirsiniz
- 2- Yerleri değiştirecek elemanlar birbirinden farklı olmalıdır
- 3- İki elementin yerleri değiştirildiğinde tüm satırlar birer yukarı kayar ve en alt satırda rastgele elementler eklenir
- 4- En üst satırda herhangi bir element ulaşlığında oyun direkt sona erer
- 5- Bir element grubunu patlatabilmek için en az 3 tane yan yana veya alta aynı elementten bulunmalıdır
- 6- Eğer hem dikey hem yatay patlama olabiliyorsa oyun en fazla element patlayabilecek olanı seçecektir
- 7- Eğer dikey ve yatay patlayabilecek element sayısı eşitse oyun yatay olanı seçer
- 8- Bir element grubunu patlatmaya karar verdığınızda bu grubun en üst veya en sol elementinin koordinatını girmeniz gerekmektedir
- 9- Yer değiştirme veya patlama işlemi seçildikten sonra koordinat yerine rakam olmayan bir karakter girmek işlemi iptal etmeyi sağlar

Oyun Modları:

- 1- Normal Mod:
Bu modda oyun matrisinin alttan $n/2$ kadar satırı rastgele elementlerle doldurulur.
Her hamleden sonra oyun ekranı temizlenir
- 2- Kontrol Mod:
Bu modda oyun matrisinin alttan $n/2$ kadar satırı ve M kadar sütununda bulunan elementler kullanıcının istenir

Oyun alanının boyutlarını ($N M$) giriniz [3-20]: 6 10
Oyun modu için 1, kontrol modu için 2 giriniz: 1

Oyunu tanıtan ekran, oyun alanı boyutunun ve oyun modunun oyuncudan istenmesi

```
Toplam yer değişikliği: 0
Toplam patlatılan element: 0

|0      1|
|1|2|3|4|5|6|7|8|9|0|
-----
01| | | | | | | | | |
-----
02| | | | | | | | | |
-----
03| | | | | | | | | |
-----
04|%|+|0|0|%|/|+|%|0|0|
-----
05|+|%|/|0|/|%|/|%|/|/
-----
06|0|*|%|*|0|*|/|%|/|


Yer değişikliği için 1
Patlama için 2
Oyundan çıkmak için -1 giriniz: |
```

Yer değişikliği ve patlatılan element sayısı ve oyun matrisi kullanıcıya gösterilir. Oyuncudan yapmak istediği işlem alınır.

```
Toplam yer değişikliği: 0
Toplam patlatılan element: 0

|0          1|
|1|2|3|4|5|6|7|8|9|0|
-----
01| | | | | | | | | |
-----
02| | | | | | | | | |
-----
03| | | | | | | | | |
-----
04|%|+|0|0|%|/|+|%|0|0|
-----
05|+|%|/|0|/|%|/|%|/|/|
-----
06|0|*|%|%|*|0|*|/|%|/|
```

Yer değişikliği için 1
Patlama için 2
Oyundan çıkmak için -1 giriniz: 1
Yerlerini değiştirmek istediğiniz iki koordinatı giriniz (format: r1,c1 r2,c2): 5,7 5,8

Yer değişikliği yapmak isteyen kullanıcı koordinatları girer.

```
Toplam yer değişikliği: 1
Toplam patlatılan element: 0

|0          1|
|1|2|3|4|5|6|7|8|9|0|
-----
01| | | | | | | | | |
-----
02| | | | | | | | | |
-----
03| | | | | | | | | |
-----
04|%|+|0|0|%|/|+|%|0|0|
-----
05|+|%|/|0|/|%|/|%|/|/|
-----
06|0|*|%|%|*|0|*|/|%|/|
```

Yer değiştirildi. Yeni satır ekleniyor
Devam etmek için Enter (Return) tuşuna basın . . .

Yer değişikliği yapılır, satırların yukarı kaydırılması ve alta yeni satır eklenmesi için kullanıcı beklenir.

```

Toplam yer değişikliği: 1
Toplam patlatılan element: 0

|0          1|
|1|2|3|4|5|6|7|8|9|0|
-----
01| | | | | | | | | |
-----
02| | | | | | | | | |
-----
03|%|+|0|0|%|/|+|%|0|0|
-----
04|+|%|/|0|/|%|%|/|/|/
-----
05|0|*|%|%|*|0|*|/|%|/
-----
06|+|0|%|0|%|+|%|*|+|+
-----
Yer değişikliği için 1
Patlama için 2
Oyundan çıkmak için -1 giriniz: |

```

Kullanıcı Enter (Return) tuşunda tıkladığında alta yeni satır eklenir ve kullanıcıya hangi işlemi yapmak istediği tekrar sorulur.

```

Toplam yer değişikliği: 1
Toplam patlatılan element: 0

|0          1|
|1|2|3|4|5|6|7|8|9|0|
-----
01| | | | | | | | | |
-----
02| | | | | | | | | |
-----
03|%|+|0|0|%|/|+|%|0|0|
-----
04|+|%|/|0|/|%|%|/|/|/
-----
05|0|*|%|%|*|0|*|/|%|/
-----
06|+|0|%|0|%|+|%|*|+|+
-----
Yer değişikliği için 1
Patlama için 2
Oyundan çıkmak için -1 giriniz: 2
Patlatmak istediğiniz bloğun sol veya üst koordinatını giriniz (format: r,c): 4,8

```

Patlatma işlemi yapmak isteyen kullanıcı koordinatları girer

```

Toplam yer değişikliği: 1
Toplam patlatılan element: 3

|0          1|
|1|2|3|4|5|6|7|8|9|0|
-----
01| | | | | | | | | |
-----
02| | | | | | | | | |
-----
03|%|+|0|0|%|/|+|%|0|0|
-----
04|+|%|/|0|/|%|%| | | |
-----
05|0|*|%|%|*|0|*|/|%|/
-----
06|+|0|%|0|%|+|%|*|+|+
-----
Patlama gerçekleşti. Yer çekimi uygulanıyor
Devam etmek için Enter (Return) tuşuna basın . . .

```

Patlama işlemi gerçekleşir. Yer çekimi uygulanması için kullanıcı beklenir.

```

Toplam yer değişikliği: 1
Toplam patlatılan element: 3

|0 1|
|1|2|3|4|5|6|7|8|9|0|
-----
01| | | | | | | | | |
-----
02| | | | | | | | | |
-----
03|%|+|0|0|%|/|+| | |
-----
04|+|%|/|0|/|%|%|0|0|
-----
05|0|*|%|%|*|0|*|/|%|/
-----
06|+|0|%|0|%|+|%|*|+|+
-----

Yer değişikliği için 1
Patlama için 2
Oyundan çıkmak için -1 giriniz: 2
Patlatmak istediğiniz bloğun sol veya üst koordinatını giriniz (format: r,c): 3,7

```

Yer çekimi uygulanır, kullanıcı bir daha patlatma yapmak ister ve koordinatları girer.

```

Toplam yer değişikliği: 1
Toplam patlatılan element: 3

|0 1|
|1|2|3|4|5|6|7|8|9|0|
-----
01| | | | | | | | | |
-----
02| | | | | | | | | |
-----
03|%|+|0|0|%|/|+| | |
-----
04|+|%|/|0|/|%|%|0|0|
-----
05|0|*|%|%|*|0|*|/|%|/
-----
06|+|0|%|0|%|+|%|*|+|+
-----

Yer değişikliği için 1
Patlama için 2
Oyundan çıkmak için -1 giriniz: 2
Patlatmak istediğiniz bloğun sol veya üst koordinatını giriniz (format: r,c): 3,7
Herhangi bir patlama olamadı!
Devam etmek için Enter (Return) tuşuna basın . . .

```

Ancak koordinatları yanlış girmiştir, girdiği koordinatta herhangi bir patlama gerçekleşmez.

```

Toplam yer değişikliği: 1
Toplam patlatılan element: 3

|0 1|
|1|2|3|4|5|6|7|8|9|0|
-----
01| | | | | | | | | |
-----
02| | | | | | | | | |
-----
03|%|+|0|0|%|/|+| | |
-----
04|+|%|/|0|/|%|%|0|0|
-----
05|0|*|%|%|*|0|*|/|%|/
-----
06|+|0|%|0|%|+|%|*|+|+
-----

Yer değişikliği için 1
Patlama için 2
Oyundan çıkmak için -1 giriniz: 2
Patlatmak istediğiniz bloğun sol veya üst koordinatını giriniz (format: r,c): 4,6

```

Bu işlemler oyun bitene veya kullanıcı ayrılanaya kadar tekrar eder.

```

Toplam yer değişikliği: 1
Toplam patlatılan element: 6

|0          1|
|1|2|3|4|5|6|7|8|9|0|
-----
01| | | | | | | | | |
-----
02| | | | | | | | | |
-----
03|%|+|0|0|%| | | | |
-----
04|+|%|/|0|/|/|+| |0|0|
-----
05|0|*|%|%|*|0|*|/|%|/
-----
06|+|0|%|0|%|+|%|*|+|+
-----

Yer değişikliği için 1
Patlama için 2
Oyundan çıkmak için -1 giriniz: 1
Yerlerini değiştirmek istediğiniz iki koordinatı giriniz (format: r1,c1 r2,c2): 4,3 4,4

```

```

Toplam yer değişikliği: 2
Toplam patlatılan element: 6

|0          1|
|1|2|3|4|5|6|7|8|9|0|
-----
01| | | | | | | | | |
-----
02|%|+|0|0|%| | | | |
-----
03|+|%|0|/|/|/|+| |0|0|
-----
04|0|*|%|%|*|0|*|/|%|/
-----
05|+|0|%|0|%|+|%|*|+|+
-----
06|+|+|%|0|/|%|*|%|/|0|


Yer değişikliği için 1
Patlama için 2
Oyundan çıkmak için -1 giriniz: 2
Patlatmak istediğiniz bloğun sol veya üst koordinatını giriniz (format: r,c): 3,4

```

```

Toplam yer değişikliği: 2
Toplam patlatılan element: 9

|0          1|
|1|2|3|4|5|6|7|8|9|0|
-----
01| | | | | | | | | |
-----
02|%|+|0| | | | | | |
-----
03|+|%|0|0|%| |+| |0|0|
-----
04|0|*|%|%|*|0|*|/|%|/
-----
05|+|0|%|0|%|+|%|*|+|+
-----
06|+|+|%|0|/|%|*|%|/|0|


Yer değişikliği için 1
Patlama için 2
Oyundan çıkmak için -1 giriniz: 1
Yerlerini değiştirmek istediğiniz iki koordinatı giriniz (format: r1,c1 r2,c2): 3,3 3,2

```

Toplam yer değişikliği: 3
Toplam patlatılan element: 9

0	1
1 2 3 4 5 6 7 8 9 0	

01 % + 0	

02 + 0 % 0 %	+ 0 0

03 0 * % % * 0 * / % /	

04 + 0 % 0 % + % * + +	

05 + + % 0 / % * % / 0	

06 + + * / * + * + % +	

Oyun en üst satırda eleman bulunduğu için sonlandırılmıştır.

En üst satırda eleman bulunduğu için sonlanan oyun.

2. Örnek

Panel De Pon oyununa hoş geldiniz!
Bu oyunun amacı karakterler en üst satıra değmeden en fazla sayıda element patlatmaya çalışmaktadır
Oyun karakterleri: [*/+%0]

Oyun kuralları:
1- İki elementi yer değiştirmek istediğiinizde sadece komşu iki elementlerin yerlerini değiştirebilirsiniz
2- Yerleri değişecek elemanlar birbirinden farklı olmalıdır
3- İki elementin yerleri değiştirildiğinde tüm satırlar birer yukarı kayar ve en alt satıra rastgele elementler eklenir
4- En üst satıra herhangi bir element ulaştığında oyun direkt sona erer
5- Bir element grubunu patlatabilmek için en az 3 tane yan yana veya alt alta aynı elementten bulunmalıdır
6- Eğer hem dikey ve yatay patlayabilecek oyun en fazla element patlayabilecek olanı seçecektir
7- Eğer dikey ve yatay patlayabilecek element sayısı eşitse oyun yatay olanı seger
8- Bir element grubunu patlatmaya karar verdığınızda bu grubun en üst veya en sol elementinin koordinatını girmeniz gerekmektedir
9- Yer değiştirme veya patlama işlemi seçildikten sonra koordinat yerine rakam olmayan bir karakter girmek işlemi iptal etmeyi sağlar

Oyun Modları:
1- Normal Mod:
 Bu modda oyun matrisinin alttan n/2 kadar satırı rastgele elementlerle doldurulur.
 Her hamleden sonra oyun ekranı temizlenir
2- Kontrol Mod:
 Bu modda oyun matrisinin alttan n/2 kadar satırı ve M kadar sütununda bulunan elementler kullanıcıdan istenir

Oyun alanının boyutlarını (N M) giriniz [3-20]: 8 5
Oyun modu için 1, kontrol modu için 2 giriniz: 2

Oyun modu için 1, kontrol modu için 2 giriniz: 2
Kontrol modunu seçtiniz 5. satırın başından 8. satırın sonuna kadarki başlangıç durumlarını girmeniz gerekmektedir.
Girebileceğiniz karakterler: [*/+%0]
5. satır, 1. sütun karakterini giriniz: +
5. satır, 2. sütun karakterini giriniz: 0
5. satır, 3. sütun karakterini giriniz: %
5. satır, 4. sütun karakterini giriniz: /
5. satır, 5. sütun karakterini giriniz: *
6. satır, 1. sütun karakterini giriniz: /
6. satır, 2. sütun karakterini giriniz: +
6. satır, 3. sütun karakterini giriniz: *
6. satır, 4. sütun karakterini giriniz: 0
6. satır, 5. sütun karakterini giriniz: *
7. satır, 1. sütun karakterini giriniz: /
7. satır, 2. sütun karakterini giriniz: %
7. satır, 3. sütun karakterini giriniz: /
7. satır, 4. sütun karakterini giriniz: *
7. satır, 5. sütun karakterini giriniz: 0
8. satır, 1. sütun karakterini giriniz: *
8. satır, 2. sütun karakterini giriniz: +
8. satır, 3. sütun karakterini giriniz: 0
8. satır, 4. sütun karakterini giriniz: +
8. satır, 5. sütun karakterini giriniz: %

```
Toplam yer değişikliği: 0
Toplam patlatılan element: 0

|0
|1|2|3|4|5|
-----
01| | | | |
-----
02| | | | |
-----
03| | | | |
-----
04| | | | |
-----
05|+|0|%|/*|
-----
06|/|+|*|0|*|
-----
07|/|%|/|*|0|
-----
08|*|+|0|+|%|
-----

Yer değişikliği için 1
Patlama için 2
Oyundan çıkmak için -1 giriniz: 1
Yerlerini değiştirmek istediğiniz iki koordinatı giriniz (format: r1,c1 r2,c2): 7,4 7,5
```

```
Toplam yer değişikliği: 1
Toplam patlatılan element: 0

|0
|1|2|3|4|5|
-----
01| | | | |
-----
02| | | | |
-----
03| | | | |
-----
04| | | | |
-----
05|+|0|%|/*|
-----
06|/|+|*|0|*|
-----
07|/|%|/|0|*|
-----
08|*|+|0|+|%|
-----

Yer değiştirildi. Yeni satır ekleniyor
Devam etmek için Enter (Return) tuşuna basın . . .
```

```
Toplam yer değişikliği: 1
Toplam patlatılan element: 0

|0
|1|2|3|4|5|
-----
01| | | | |
-----
02| | | | |
-----
03| | | | |
-----
04|+|0|%|/*|
-----
05|/|+|*|0|*|
-----
06|/|%|/|0|*|
-----
07|*|+|0|+|%|
-----
08|0|/|%|*|%|
-----

Yer değişikliği için 1
Patlama için 2
Oyundan çıkmak için -1 giriniz: 2
Patlatmak istediğiniz bloğun sol veya üst koordinatını giriniz (format: r,c): 4,5
```

```
Toplam yer değişikliği: 1
Toplam patlatılan element: 3

|0
|1|2|3|4|5|
-----
01| | | | |
-----
02| | | | |
-----
03| | | | |
-----
04|+|0|%|/| |
-----
05|/|+|*|0| |
-----
06|/|%|/|0| |
-----
07|*|+|0|+|%|
-----
08|0|/|%|*|%|
-----
Yer değişikliği için 1
Patlama için 2
Oyundan çıkmak için -1 giriniz: 1
Yerlerini değiştirmek istediğiniz iki koordinatı giriniz (format: r1,c1 r2,c2): 7,3 7,4
```

```
Toplam yer değişikliği: 2
Toplam patlatılan element: 3

|0
|1|2|3|4|5|
-----
01| | | | |
-----
02| | | | |
-----
03| | | | |
-----
04|+|0|%|/| |
-----
05|/|+|*|0| |
-----
06|/|%|/|0| |
-----
07|*|+|0|%|
-----
08|0|/|%|*|%|
-----
Yer değiştirildi. Yeni satır ekleniyor
Devam etmek için Enter (Return) tuşuna basın . . .
```

```
Toplam yer değişikliği: 2
Toplam patlatılan element: 3

|0
|1|2|3|4|5|
-----
01| | | | |
-----
02| | | | |
-----
03|+|0|%|/| |
-----
04|/|+|*|0| |
-----
05|/|%|/|0| |
-----
06|*|+|0|%|
-----
07|0|/|%|*|%|
-----
08|*|*|+|%|*|
-----
Yer değişikliği için 1
Patlama için 2
Oyundan çıkmak için -1 giriniz: 1
Yerlerini değiştirmek istediğiniz iki koordinatı giriniz (format: r1,c1 r2,c2): 7,4 7,5
```

```
Toplam yer değişikliği: 3
Toplam patlatılan element: 3

|0
|1|2|3|4|5|
-----
01| | | | |
-----
02| | | | |
-----
03|+|0%|/| |
-----
04|/|+|*|0| |
-----
05|/|%|/|0| |
-----
06|*|+|+|0|%|
-----
07|0|/|%|%|*|
-----
08|*|*|+|%|*|
Yer değiştirildi. Yeni satır ekleniyor
Devam etmek için Enter (Return) tuşuna basın . . .
```

```
Toplam yer değişikliği: 3
Toplam patlatılan element: 3

|0
|1|2|3|4|5|
-----
01| | | | |
-----
02|+|0%|/| |
-----
03|/|+|*|0| |
-----
04|/|%|/|0| |
-----
05|*|+|+|0|%|
-----
06|0|/|%|%|*|
-----
07|*|*|+|%|*|
-----
08|*|*|*|/|*|
Yer değişikliği için 1
Patlama için 2
Oyundan çıkmak için -1 giriniz: 2
Patlatmak istediğiniz bloğun sol veya üst koordinatını giriniz (format: r,c): 3,4
```

```
Toplam yer değişikliği: 3
Toplam patlatılan element: 6

|0
|1|2|3|4|5|
-----
01| | | | |
-----
02|+|0%| | |
-----
03|/|+|*| | |
-----
04|/|%|/| | |
-----
05|*|+|+|/|%|
-----
06|0|/|%|%|*|
-----
07|*|*|+|%|*|
-----
08|*|*|*|/|*|
Yer değişikliği için 1
Patlama için 2
Oyundan çıkmak için -1 giriniz: 1
Yerlerini değiştirmek istediğiniz iki koordinatı giriniz (format: r1,c1 r2,c2): 8,4 8,5
```

Toplam yer değişikliği: 4
Toplam patlatılan element: 6

0	
1 2 3 4 5	

01	

02	+ 0 %

03	/ + *

04	/ % /

05	* + / / %

06	0 / % % *

07	* * + % *

08	* * * * /

Yer değiştirildi. Yeni satır ekleniyor
Devam etmek için Enter (Return) tuşuna basın . . .

Toplam yer değişikliği: 4
Toplam patlatılan element: 6

0	
1 2 3 4 5	

01	+ 0 %

02	/ + *

03	/ % /

04	* + / / %

05	0 / % % *

06	* * + % *

07	* * * * /

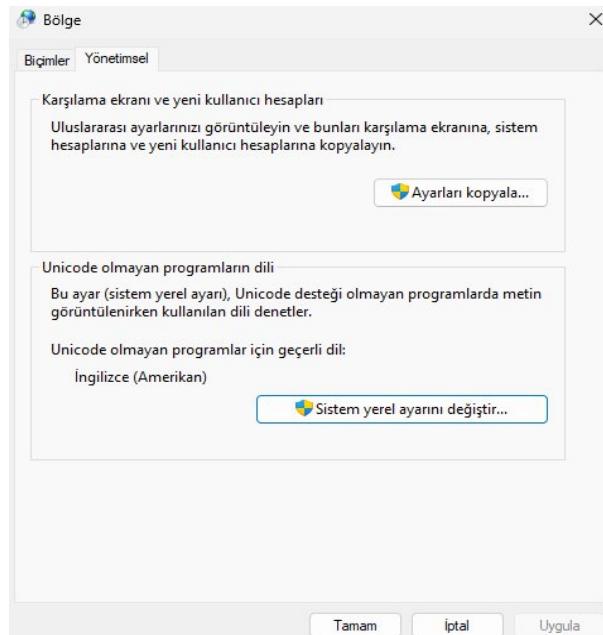
08	% * / / %

Oyun en üst satırda eleman bulunduğu için sonlandırılmıştır.

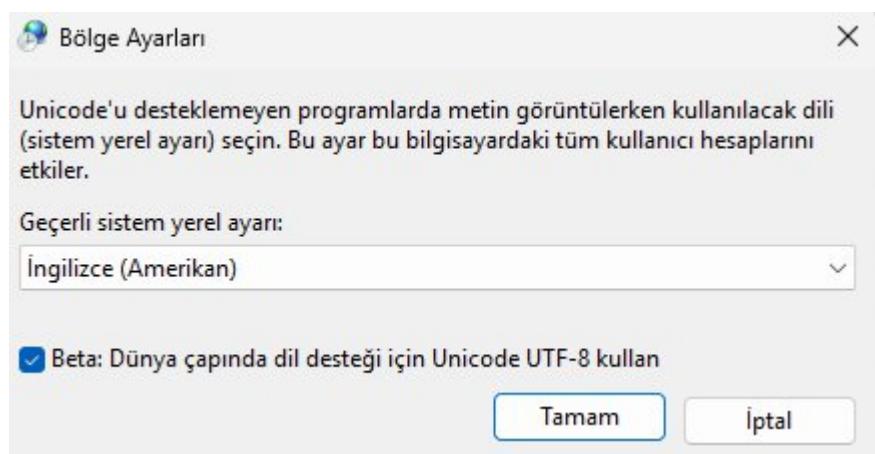
Hata Giderme

Eğer Dev-C++ (veya UTF-8 desteklemeyen başka bir IDE) ve terminal'de Türkçe karakterler bozuk gözükmüyorsa şu adımlar izlenerek bu sorun düzeltilebilir:

- 1) Klavyeden Win+R tuş kombinasyonuna basın, “intl.cpl” yazıp çalıştırın
- 2) “Yönetimsel” sekmesine gelip ve “Sistem yerel ayarını değiştir...” butonuna tıklayın.



- 3) “Beta: Dünya çapında dil desteği için Unicode UTF-8 kullan” butonunu işaretleyip “Tamam” butonuna tıklayın.



- 4) Windows'u yeniden başlatın.

```

145 void print_splash_screen(void)
146 {
147     /* stringleri parçalara ayırdık çünkü bu kadar uzun bir string c89 (ansi c) tarafından desteklenmez
148     puts(
149         "Panel De Pon oyununa hoş geldiniz!\n"
150         "Bu oyunun amacı karakterler en üst satırda deðmeden en fazla sayıda element patlatmaya ñsalatmak"
151         "Oyun karakterleri: [*/+%0]\n"
152         "\n"
153         "Oyun kuralları: \n"
154         " 1- ïki elementi yer değiştirmek istediginizde sadece komâyu iki elementlerin yerlerini deðiþir"
155         " 2- Yerleri değiþecek elemanlar birbirinden farklı olmalıdır\n"
156         " 3- ïki elementin yerleri deðiþirildiðinde tüm satırlar birer yukarıda kayar ve en alt satır"
157     );
158     puts(
159         " 4- En üst satırda herhangi bir element ulaþtığında oyuna direkt sona erer\n"
160         " 5- Bir element grubunu patlatabilmek için en az 3 tanesi yan yana veya alt alta aynı elementten olabilir"
161         " 6- Eğer hem dikey hem yatay patlama olabiliyorsa oyuna fazla element patlayabilecek olan sayı"
162     );
163     puts(
164         " 7- Eğer dikey ve yatay patlayabilecek element sayısı 8'den fazlaysa oyuna yatay olanları seðit"
165         "\n"
166         "Oyun Modları: \n"
167         " 1- Normal Mod...\n"

```

Oyun kuralları:

- 1- ïki elementi yer değiştirmek istediginizde
- 2- Yerleri değiþecek elemanlar birbirinden farklı olmalıdır
- 3- ïki elementin yerleri deðiþirildiðinde tüm satırlar birer yukarıda kayar ve en alt satırda sona erer
- 4- En üst satırda herhangi bir element ulaþtığında oyuna direkt sona erer
- 5- Bir element grubunu patlatabilmek için en az 3 tanesi yan yana veya alt alta aynı elementten olabilir
- 6- Eğer hem dikey hem yatay patlama olabiliyorsa oyuna fazla element patlayabilecek olan sayı 8'den fazlaysa oyuna yatay olanları seðit
- 7- Eğer dikey ve yatay patlayabilecek element sayısı 8'den fazlaysa oyuna yatay olanları seðit
- 8- Bir element grubunu patlatmaya karar verdiðinizde bu grubun en üst veya en sol elementinin hedefi olmalıdır
- 9- Yer deðiþirme veya patlama işlemi seðildikten sonra koordinat yerine rakam olmayan bir karakter kullanılamaz

Ayarı değiştirmeden önce

```

145 void print_splash_screen(void)
146 {
147     /* stringleri parçalara ayırdık çünkü bu kadar uzun bir string
148     puts(
149         "Panel De Pon oyununa hoş geldiniz!\n"
150         "Bu oyunun amacı karakterler en üst satırda deðmeden en"
151         "Oyun karakterleri: [*/+%0]\n"
152         "\n"
153         "Oyun kuralları: \n"
154         " 1- İki elementi yer değiştirmek istediginizde sadece"
155         " 2- Yerleri değiþecek elemanlar birbirinden farklı olmalıdır"
156         " 3- İki elementin yerleri değiştirildiðinde tüm satırlar"
157     );
158     puts(
159         " 4- En üst satırda herhangi bir element ulaþtığında oyuna"
160         " 5- Bir element grubunu patlatabilmek için en az 3 tanesi"
161         " 6- Eğer hem dikey hem yatay patlama olabiliyorsa oyuna"
162         " 7- Eğer dikey ve yatay patlayabilecek element sayısı"
163     );

```

Panel De Pon oyununa hoş geldiniz!
 Bu oyunun amacı karakterler en üst satırda deðmeden en
 Oyun karakterleri: [*/+%0]

Oyun kuralları:

- 1- İki elementi yer değiştirmek istediginizde sadece
- 2- Yerleri değiþecek elemanlar birbirinden farklı olmalıdır
- 3- İki elementin yerleri değiştirildiðinde tüm satırlar birer yukarıda kayar ve en alt satırda sona erer
- 4- En üst satırda herhangi bir element ulaþtığında oyuna direkt sona erer
- 5- Bir element grubunu patlatabilmek için en az 3 tanesi yan yana veya alt alta aynı elementten olabilir
- 6- Eğer hem dikey hem yatay patlama olabiliyorsa oyuna fazla element patlayabilecek olan sayı 8'den fazlaysa oyuna yatay olanları seðit
- 7- Eğer dikey ve yatay patlayabilecek element sayısı 8'den fazlaysa oyuna yatay olanları seðit
- 8- Bir element grubunu patlatmaya karar verdiðinizde bu grubun en üst veya en sol elementinin hedefi olmalıdır

Ayarı değiştirdikten sonra