

به نام خدا

MineSweeper

پارسا صمدنژاد

نحوه کارکرد برنامه :

ابتدا آرایه 2 بعدی را به عنوان صفحه بازی در نظر میگیریم، سپس تا وقتی که 40 خانه فلگ دار یا خالی باقی مانده باشد پیش میرویم و از کاربر ورودی میگیریم.

اگر کاربر خانه ای که در آن بمب باشد را انتخاب کند، بازی تمام شده و صفحه بازی در آخرین مرحله به اضافه تمام بمب ها را به او نشان میدهیم.

اگر به 40 خانه خالی یا فلگ دار برسیم ، ینی کاربر تمام خانه هایی که بمب نبوده را تشخیص داده و برنده میشود.

دستورات بازی به صورت :

دستور انتخاب : cho

دستور پرچمگذاری : fla

دستور پرچم برداری : ufl

دستور خروج : ext

بعد از هر کدام از دستورات زیر رشته ای به طول حداکثر 5 برای مختصات نقطه مورد نظر میگیریم و برای تبدیل رشته ورودی به اعداد که از آنها استفاده کنیم از توابع :

(رشته مختصات) int inputX

(رشته مختصات) int inputY

برای بمب ها به صورت رندوم از تابع :

(تعداد بمب, آرایه ای برای ذخیره شماره بمب ها) randMine

استفاده میکنیم که 40 مقدار غیر تکراری تولید میکند و هر مقدار نشان دهنده شماره خانه بمب است که از سمت چپ بالا

خانه ها به ترتیب سطر از 0 تا 14*30 منهای یک شماره گذاری میکنیم.

تمام خانه های آرایه صفحه بازی را در ابتدا به وسیله ی

void defineTable(آرایه)

به # تبدیل میکنیم.

سپس تا وقتی که کاربر برنده نشده لوپ زیر را اجرا میکنیم:

دستور و مختصات را از کار بر میگیریم

1. اگر دستور پرچم گذاری بود به وسیله ی :

void setFlag(char table[M][N],int,int)

اگر خانه # باشد به پرچم تبدیل میکنیم.

2. اگر دستور پرچم برداری بود به وسیله ی:

void unflag(char table[M][N],int,int)

اگر خانه پرچم بود به # تبدیل میکنیم.

3. اگر دستور انتخاب بود ابتدا چک میکنیم که آیا خانه خواسته شده دارای بمب است یا نه:

به وسیله ی تابع :

checker(mines,Mine,x,y)

مختصات خانه و تعداد بمب ها(40) و آرایه بمب ها را با تابع می‌دهیم اگر بمب بود 1 و اگر نبود 0 را باز میگرداند

اگر خانه دارای بمب بود صفحه بازی پاک میشود و بمب ها به وسیله ی تابع:

defineMines(table,mines,Mine)

بر روی آرایه صفحه بازی قرار میگیرند و سپس آنرا به همراه پیام باخت نمایش داده و برنامه را تمام میکنیم .

اگر خانه دارای بمب نبود مختصات خانه را به همراه آرایه صفحه بازی و تعداد بمب ها و آرایه دارای بمب ها و آرایه ای که به آن :

char tableCheck[M][N]

میگوییم را به تابعی به نام :

```
void sweeper(char table[M][N],char tableCheck[M][N],int mines[],int,int ,int )
```

پاس میدهیم.

کاری که این تابع انجام میدهد این است که :

اگر خانه در کنارش بمب باشد تعداد بمب های خانه اطرافش را با # جایگزین میکند و تابع تمام میشود

اگر در خانه های کنارش بمب نبود به جای # یک کاراکتر جای خالی میگذارد و سپس تمام خانه های اطرافش را که در صفحه بازی هستند به تابع پاس داده میشوند و این خانه در آرایه

tableCheck

از # به 0 تبدیل میشود که بار دیگر که این تابع بر روی مختصات دیگری صدا زده شد چک بکند که آیا مقدار آن در

tableCheck

برابر 0 است یا # اگر 0 بود یعنی قبلا این تابع بروی آن صدا زده شده و دیگر هیچ کاری نکند

و این صدا زدن تابع تا جایی ادامه پیدا میکند که یا خانه ای قبلا چک شده باشد یا در کنارش بمب باشد

sweeper پس از این که کار با تابع

در این مرحله تمام شد تمام خانه های آرایه ی

tableCheck

را به # تبدیل میکنیم تا باز هم این روند در دور های بعدی به درستی انجام بگیرد.

4. اگر دستور خروج وارد شود در تابع اصلی مقدار 0 بازگردانده میشود و برنامه تمام میشود.

نحوه کارکرد بقیه ی توابع :

void cls():

صفحه ترمینال را پاک میکند.

void winner():

پس از اتمام بازی و برنده شدن شخص پیامی را نشان میدهد.

void loser():

پس از اتمام بازی و باخت کاربر پیام باخت را نشان میدهد.

int winCheck(char table[M][N]):

عددی را باز میگرداند که نشان دهنده ی تعداد خانه هایی اس که بازیکن میتواند آن هارا انتخاب کند که اگر به 40 برسد تمام خانه های باقی مانده مین هستند و بازی کن برنده میشود.

int inputX(char xY[]):

رشته وارد شده توسط کاربر را گرفته و تعداد ارقام وارد شده تا قبل از "،" را بدست آورده و باتوجه به مکان هر رقم قبل از ، مقدار ایکس را به عدد صحیح تبدیل کرده و باز میگرداند.

int inputY(char xY[]):

رشته وارد شده توسط کاربر را گرفته و تعداد ارقام وارد شده بعد از "،" را بدست آورده و باتوجه به مکان هر رقم بعد از ، مقدار ایگرگ را به عدد صحیح تبدیل کرده و باز میگرداند.

int checker(int mines[],int ,int,int):

تابعی است که با وارد کردن مختصات و آرایه ای که 40 مین در آن ذخیره شده مقدار یک یا صفر را که نشان دهنده وجود مین در آن مختصات است یا نه را باز میگرداند.

void defineTable(char table[M][N]):

تابعی است که با پاس دادن آرایه به آن ، تمام کاراکتر های آرایه را به # تبدیل میکند.

void printTable(char table[M][N]):

تابعی است که با پاس دادن آرایه ای به آن ، آن را چاپ میکند.

void randMine(int mines[],int size):

با پاس دادن آرایه ای و مقدار مین ها به صورت رندوم به مقداری که میخواهیم رندوم بدست آورده و در آرایه ذخیره

میکند.

*****توضیحات بیشتر به صورت کامنت در کد نوشته شده است**