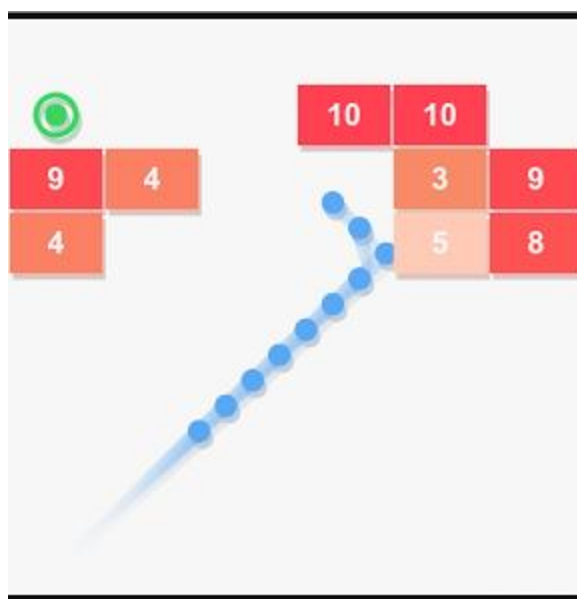


هدف این پروژه طراحی تغییریافته بازی "Bricks breaker" می

توضیحات کلی:

در هر مرحله از این بازی، روی صفحه بازی که یک مستطیل به $3N \times 2N$ یک ردیف شامل تعدادی آجر و گاهای بمب صفحه مرحله قبل اضافه می شود کاربر باید با نشانه گیری P به سمت این آجرها تیراندازی کند. هر آجر دارای مقاومت R_b در برابر تیراندازی است یعنی باید R_b تا تیر به آن برخورد کند تا از صفحه بازی حذف شود یعنی هر تیر از مقاومت آجر یک واحد کم می کند و آجر با مقاومت صفر از صفحه بازی حذف می شود و هر آجری که از صفحه حذف شود یک واحد به امتیاز کاربر (S_p) اضافه می شود. k تیر به بمب برخورد کند از مقاومت تمام آجرهای مجاور راسی آن k واحد کم می شود. بازی زمانی به پایان می رسد که آجری پایین صفحه بازی برخورد کند.



1. بخش اصلی:

این بخش، ابتدای بازی از کاربر ابعاد صفحه بازی (N)
بگیرید. نقطه پرتاب تیر را فرض کرده $P = (1, N)$

i ام تمام آجرهای باقی مانده از مرحله قبل یک ردیف
پایین انتقال دهید و به طوری که سطح پایین صفحه بازی
نزدیک تر شوند و در ردیف $3N$ تعدادی رندوم T_b
($1 \leq T_b < 2N$) با مقاومت های رندوم R_i
رندوم بین 0 تا 1 در نظر بگیرید، اگر $g + (T_b/2N)$ بیشتر از 1 بود یک
خانه خالی در این ردیف به صورت رندوم بمب خواهد بود.
به کاربر تمام مختصات آجرها به همراه مقاومت آن ها و امتیاز
کاربر را اطلاع می دهید سپس از کاربر یک زاویه
($0 < \phi < 180$) بگیرید.

```
Level 2:
  Bricks:
      Row 6:
      2 0 1 0
      Row 5:
      1 1 0 1
  phi:
-----
```

(به طور مثال کاربر ϕ 120 درجه وارد می کند.)

$P=(1,N)$ S_p تا تیر با زاویه (جهت مثلثاتی)
 پرتاب می شود، اگر تیری به آجری برخورد کند از مقاومت آن یک
 واحد کم می شود، دقت کنید تیرها بازتاب نمی شوند و طبعاً از
 آجرها عبور نمی کنند. (می توانید تیرها را با فاصله زمانی
 کوتاهی پشت سر هم در نظر بگیرید.) اگر آجر یا آجرهایی
 مقاومتشان صفر شد به کاربر امتیاز جدیدش را اطلاع دهید.

```
Level 2:
    Bricks:
        Row 6:
        2 0 1 0
        Row 5:
        1 1 0 1
    phi:120
Score: 2
```

2. *load save*:

در ابتدای برنامه از بازیکن پرسیده می شود که آیا می
 خواهد بازی جدیدی را شروع کند (new game) یا بازی
 ذخیره شده ای را بارگذاری کند (load game). پایان هر
 بازی نیز بازیکن با دادن ورودی ای (s) می تواند
 بازی خود را ذخیره کند. در هنگام ذخیره سازی همه ی
 اطلاعات لازم باید ذخیره شود به صورتی که در صورتی
 بارگذاری آن، بازی بتواند بدون هیچ مشکلی روند خود را
 ادامه دهد. بازی می تواند با نام دلخواه که کاربر وارد می
 کند ذخیره شود، ولی اگر نام بازی های ذخیره شده به

#shatnard، که # یک شماره است، ذخیره شود،
نمره ی امتیازی خواهد داشت. منظور از # این است که در
ابتدا این عدد 1 است، ولی در ادامه در صورت وجود داشتن
1، با شماره ی 2 2 و 3 ذخیره شود.
بدیهی است پس از ذخیره کردن این نام باید برای کاربر
نمایش داده شود. .

3. گرافیک:

برای گرافیک می توانید کتابخانه های *graphics.h* یا
QT کنید. این کافی که ی
بازی گرافیک نمایش بازیکن می طریق
کنسول، ورودی دهد. صورتی که ی بی نیاز
کنسول ی امتیازی خواهد . طراحی
تصویری در کنسول هم شامل بخشی از نمره خواهد شد.

4. نقطه پرتاب تیر متغیر:

در هر مرحله از کاربر
Sp تیز از نقطه پرتاب با زاویه پرتاب کنید.
پرتاب تیر را بگیرید.

5. تیر با قابلیت بازتاب:

اگر تیری به دیوار یا آجر برخورد کند بازتاب منظم می کند.
(بازتاب منظم بازتابی است با هر زاویه ای
کند با همان زاویه از جسم دور می شود مانند بازتاب در
آینه)