



## Lonely Cells

### فهرست مطالب

1	مقدمه
2	روند کلی بازی
2	نقشه
2	بلوک‌ها
3	سلول‌ها
4	فایل نقشه
5	مراحل اجرای برنامه
8	نکات پیاده‌سازی
9	قسمت‌های امتیازی
10	قدردانی

### مقدمه

هدف از این پروژه، توسعه یک بازی یک یا دو نفره است که جزئیات آن در ادامه آمده است. بازی زندگی چند تک سلولی را در یک جدول دنبال می‌کند، زندگی این تک سلولی‌ها بسیار یکنواخت و خسته کننده است 😞. آن‌ها می‌توانند در خانه‌های بی‌روح این جدول حرکت کنند، انرژی کسب کنند و تقسیم شوند.

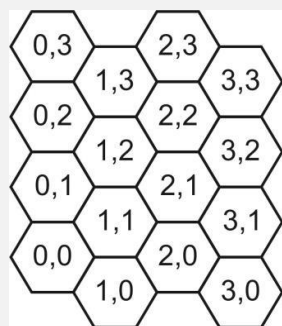
خروجی مورد انتظار از پروژه در بخش "مراحل اجرای برنامه" توضیح داده شده است.

## روند کلی بازی

بازی در نقشه‌ای که از بلوک (خانه)های ۶ ضلعی تشکیل شده است، انجام می‌شود. در ابتدای بازی، از کاربر تعداد سلول‌های مورد نیاز پرسیده می‌شود و آن‌ها به صورت تصادفی در بلوک‌های نقشه قرار داده می‌شوند. در ادامه این سلول‌ها با دریافت دستوراتی که از کاربر دریافت می‌کنند اعمالی را انجام می‌دهند.

اجرای بازی به صورت نوبتی است و هر سلول در هر نوبت می‌تواند یکی از اعمال ممکن در بازی را انجام دهد که شامل حرکت به بلوک‌های مجاور، جمع آوری منبع و یا تقسیم سلول است. البته دو عمل جمع آوری منبع و تقسیم سلول فقط می‌توانند در خانه‌های خاصی از نقشه انجام شوند. در انتهای بازی مجموع انرژی کسب شده توسط سلول‌های کاربر محاسبه شده و به عنوان امتیاز کاربر در نظر گرفته می‌شود. بازی را باید به دو صورت یک نفره و دو نفره پیاده‌سازی کنید. در ادامه جزئیات بازی به تفصیل توضیح داده می‌شود.

## نقشه



مختصات نقشه بازی به صورت شکل زیر است که جنوب غربی ترین نقطه‌ی نقشه، مختصات (۰،۰) دارد. مولفه اول نشان دهنده‌ی مولفه‌ی  $x$  و مولفه دوم نشان دهنده‌ی مولفه‌ی  $y$  است. نکته قابل توجه درباره نقشه این است که این نقشه (برخلاف ظاهرش) یک نقشه مربعی معمولی است. و تنها تفاوت آن تفاوت در تعداد خانه‌های مجاور است. بنابراین در کار با فایل و ذخیره کردن و خواندن نقشه به این نکته توجه داشته باشید. جزئیات بیشتر در ادامه توضیح داده خواهد شد.

## بلوک‌ها

بلوک‌ها به یکی از انواع زیر دیده می‌شوند:

- خانه منبع: خانه ای است که منبع در آن وجود دارد. اگر در نوبتی که یک سلول روی این خانه است دستور جمع آوری منبع به آن سلول داده شود مقدار مشخصی منبع را کسب می‌کند. در نظر داشته باشید که مقدار منبع در این خانه‌ها محدود و برابر با مقدار ۱۰۰ است.
- خانه تقسیم: خانه ای که عمل تقسیم (دو تا شدن) میتواند در آن انجام شود. اگر در نوبتی که یک سلول روی این خانه است دستور تقسیم به آن سلول داده شود در صورت وجود شرایط لازم سلول تقسیم شده و سلول جدید به خانه مجاور میرود.

- خانه غیر قابل عبور: خانه‌هایی هستند که رفتن به آنها ناممکن است.
- خانه معمولی: خانه‌های معمولی نقشه هستند که ویژگی‌های هیچ کدام از انواع فوق را ندارند.

## سلول‌ها

هر کاربر دارای تعدادی سلول است که در طول بازی سعی می‌کند مجموع انرژی آن‌ها را بیشینه کند. در این مستند همیشه منظور از سلول موجودات سلولی است و اشاره به خانه‌های نقشه ندارد. در ادامه ویژگی‌های این سلول‌ها را توضیح می‌دهیم. هر سلول دارای سطح انرژی است که میزان گلوکز ذخیره شده در سلول را نشان می‌دهد، این مقدار حداکثر برابر ۱۰۰ واحد خواهد بود. در هر نوبت کاربر می‌تواند به هر سلول یک دستور دهد و آن سلول در صورتی که شرایط لازم برای اجرای دستور وجود داشته باشد آن را انجام می‌دهد. دستورات ممکن در ادامه توضیح داده می‌شوند.

• حرکت: دستور حرکت نیاز به یک جهت دارد که مشخص کننده‌ی یک خانه‌ی همسایه‌ی خانه‌ی محل حضور سلول است. در مجموع شش جهت شمال، شمال شرقی، جنوب شرقی، جنوب، جنوب غربی و شمال غربی وجود دارد.

○ حرکت در صورتی انجام می‌شود که بلوک مقصد از نوع غیر قابل عبور نباشد و خالی باشد (دقت کنید که حرکت فقط می‌تواند به بلوک‌های نقشه انجام شود و نمی‌توان از نقشه خارج شد!)

• تقسیم: با ارسال این دستور در صورت وجود شرایط زیر، سلولی که دستور را دریافت کرده به دو سلول تبدیل می‌شود.

○ عمل تقسیم سلولی تنها در بلوک‌های تقسیم، قابل انجام است و در بلوک‌های دیگر، در صورت ارسال این دستور، اتفاقی نمی‌افتد.

○ عمل تقسیم، تنها در صورتی انجام می‌شود که در هنگام صدور دستور، سلول جدید بتواند به یکی از خانه‌های همسایه برود که شرایط آن مانند شرایط حرکت سلول است. سلول جدید به صورت تصادفی به یکی از خانه‌هایی که شرایط حرکت به آنها وجود دارد، می‌رود.

○ سطح انرژی سلول در هنگام صدور دستور تقسیم باید بالاتر از ۸۰ باشد و اگر کمتر باشد اتفاقی نخواهد افتاد.

○ پس از انجام عمل تقسیم، دو سلول با سطح انرژی ۴۰ خواهیم داشت. در نظر داشته باشید که اگر سطح انرژی سلول از ۸۰ هم بیشتر باشد باز ما دو سلول با سطح انرژی ۴۰ خواهیم داشت.

• جمع آوری منبع: در صورتی که سلول در یک خانه‌ی منبع باشد و این دستور داده شود، مقداری انرژی از خانه‌ی منبع کم شده و به سلول اضافه می‌شود. در صورتی که مقدار انرژی خانه منبع بیشتر یا مساوی ۱۵ باشد مقدار انرژی که جمع آوری می‌شود برابر ۱۵ است و در غیر این صورت به اندازه‌ی انرژی موجود در خانه‌ی منبع است که بعد از آن انرژی آن خانه منبع به انتها می‌رسد.

هر سلول دارای یک نام است که می‌توانید آن را به صورت تصادفی ساخته یا آن را از کاربر بپرسید. زمانی که یک سلول تقسیم می‌شود سلول اولیه از **بین رفته** (از لیست حذف شود) و نیاز است که دو سلول با نام‌های جدید ایجاد کنید.

برای تولید اسم‌ها (رشته‌های) تصادفی راه‌های زیادی دارید یکی از آن‌ها:

```
char *rand_string(size_t size) {
    // reserves a place for NULL
    char *str = malloc((size + 1) * sizeof(char));

    const char charset[] =
"abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ";

    for (size_t i = 0; i < size; i++) {
        int key = rand() % (sizeof(charset) / sizeof(char) - 1);
        str[i] = charset[key];
    }
    str[size] = '\0';

    return str;
}

int main() {
    srand(time(NULL));

    char *str = rand_string(5);
    printf("rand string[5]: %s\n", str);
    free(str);
}
```

## فایل نقشه

نقشه‌ی این بازی به صورت یک فایل باینری به همراه این مستند در اختیار شما قرار گرفته است. توجه کنید در زمان تحویل پروژه فایل دیگری با همین فرمت در اختیار شما قرار داده می‌شود. فرمت این نقشه به صورت زیر است.

همانطور که بیان شد، شما یک جدول مربعی  $n \times n$  در اختیار دارید که هر یک از خانه‌های آن چهار حالت دارند که آن‌ها را با اعداد زیر نمایش می‌دهیم:

```
enum blocks {  
    ENERGY = 1,  
    MITOSIS,  
    FORBIDDEN,  
    NORMAL  
};
```

یا به عبارت دیگر:

Energy	1
Mitosis	2
Forbidden	3
Normal	4

در فایلی که به شما داده می‌شود در ابتدا یک عدد صحیح بدون علامت ۳۲ بیتی می‌خوانید که  $n$  می‌باشد. در ادامه  $n \times n$  کاراکتر بدون علامت ۸ بیتی دارید که هر یک وضعیت یک خانه را نمایش می‌دهند. در نظر داشته باشید این اعداد به صورت Row Major آرایش شده‌اند.

در ادامه محتوای باینری و متنی متناظر با آن را برای یک فایل نقشه ۲ در ۲ می‌بینید:

```
00000000: 0200 0000 3132 3334      ....1234
```

## مراحل اجرای برنامه

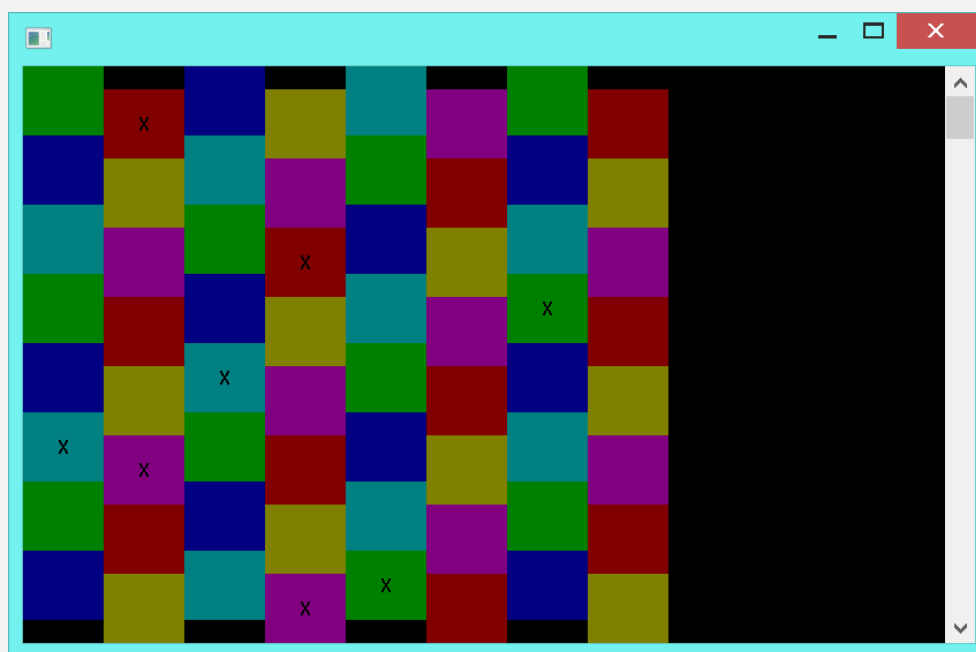
۱- در ابتدای بازی منوی زیر به کاربر نشان داده می‌شود. کاربر می‌تواند هر یک از منوهای ۱ و ۲ و ۳ و ۴ را انتخاب کند و متناسب با منوی انتخابی به مرحله ی بعد برود. جزئیات عملکرد هر منو در ادامه شرح داده شده است.

```
[1]Load  
[2]New single player game  
[3]New Multiplayer game  
[4]Exit
```

[ 5 ]

۲- اگر در مرحله‌ی قبل منوی ۲ و یا ۳ انتخاب شده باشد، در این مرحله تعداد سلول‌های مورد نیاز از کاربر پرسیده می‌شود و آن‌ها به صورت تصادفی در بلوک‌های نقشه قرار داده می‌شوند.

۳- در ادامه نقشه‌ی بازی که در فایل ذخیره شده است و به همراه این مستند در اختیار شما قرار گرفته است، به کاربر نشان داده می‌شود. توجه کنید علاوه بر خانه‌های نقشه، سلول‌های کاربر که در مرحله قبلی گرفته شدند نیز باید در نقشه نشان داده شود. نمای نمونه از نقشه می‌تواند به شکل زیر باشد. در این نمونه سلول‌ها در خانه‌هایی که با X نشان داده شده‌اند قرار دارند و همانطور که مشاهده می‌شود هر خانه از نقشه، ۶ خانه‌ی همسایه دارد. در نظر داشته باشید که در حالت دو نفره می‌بایست مشخص کنید که هر سلول متعلق به کدام کاربر می‌باشد.



۴- در این زمان از کاربر خواسته می‌شود که یکی از سلول‌های خودش را انتخاب کند:

Please choose one of your cells:

[1] Parham (0, 2)

[2] ...

در نمونه فوق Parham نام سلول بوده و (0, 2) مختصات آن می‌باشند.

۵- در این مرحله منوی اصلی زیر به کاربر نشان داده می‌شود.

[6]

```
[1]Move  
[2]Split a cell  
[3]Boost energy  
[4]Save  
[5]Exit
```

کاربر می تواند هر یک از منوهای ۱ و ۲ و ۳ و ۴ را انتخاب کند و بعد از اتمام کار آن منو، دوباره باید به این منوی اصلی برگردد. در نظر داشته باشید که منو ۵ بازی را خاتمه می دهد. جزئیات عملکرد هر منو در ادامه شرح داده شده است.

#### ۶- منوی حرکت

در این مرحله منوی زیر به کاربر نشان داده می شود.

```
[1]North  
[2]South  
[3]Northeast  
[4]Northwest  
[5]Southeast  
[6]Southwest
```

زمانی که کاربر حرکت خود را انتخاب کرد، شما وظیفه دارید که جدول را به روز رسانی کنید.

#### ۷- منوی تقسیم و منوی جمع آوری منبع

دستورات تقسیم و جمع آوری منبع طبق تعریفی که در بخش قبلی ارائه شد اجرا می شوند. نقشه ی بازی مجدداً به کاربر نشان داده می شود و سپس منوی اصلی به کاربر نشان داده می شود.

#### ۸- منوی ذخیره ی بازی

با انتخاب این منو وضعیت نقشه و همچنین وضعیت بازیکن(ها) در فایل های جداگانه ای ذخیره می شود.

[ 7 ]

- تمامی اطلاعات مربوط به وضعیت بازیکن(ها) که برای شروع مجدد بازی لازم است باید ذخیره شود (تعداد سلول ها، مکان قرارگیری هر سلول، سطح انرژی هر سلول).

- نقشه باید به همان فرمتی که در اختیار شما قرار گرفته است ذخیره شود تا در دفعات بعدی اجرا قابل استفاده باشد.

۹- منوی load

با انتخاب این منو تمامی اطلاعات بازیکن(ها) و نقشه از فایل های مربوطه خوانده می شود. ادامه بازی باید براساس وضعیت های خوانده شده صورت گیرد. وضعیت فعلی بازی به کل پاک می شود.

۹- منوی خروج

با انتخاب این منو بازیکن(ها) از برنامه خارج می شوند و بازی خاتمه می یابد.

## نکات پیاده سازی

- برنامه خود را به صورت مناسب به توابع مختلفی بشکنید (پیاده سازی این برنامه تنها با یک تابع main قابل قبول نیست).

- باید دقت کنید که این بازی دو حالت دارد، یک حالت یک نفره و یک حالت دو نفره. این نکته در ابتدای شروع برنامه از کاربر پرسیده می شود.

- نقشه ی این بازی باید به صورت زیبا و کاربر پسند در محیط کنسول نمایش داده شود. توجه کنید که نمای ارائه شده در این مستند تنها نمونه ای از پیاده سازی نقشه است و شما می توانید نقشه را به هر صورتی که مفهوم داشتن شش همسایه را برساند، نمایش دهید و نیازی به نمایش با رنگ نمی باشد.

- سلول ها می بایست در یک لیست پیوندی نگهداری شوند. در صورتی که آن ها را با استفاده از آرایه پیاده سازی کنید بخشی از نمره را از دست خواهید داد.

- در حالت دو نفره منو عملیات سلول برای هر کاربر به نوبت نمایش داده می شود و هر کاربر در نوبت خود می تواند روی یک سلول خودش عملیات انجام دهد.

- پیش از شروع پیاده سازی حتما یکبار تمامی جزئیات پروژه با دقت بخوانید و در صورت نیاز ابهامات را با تدریساران در میان بگذارید.



## قسمت‌های امتیازی

- طراحی یک Map Editor برای بازی به این صورت که کاربر بتواند نقشه دلخواه خود را طراحی کند و آن را در یک فایل ذخیره کند. مثلاً بتواند محل بلوک‌های منبع و محل بلوک‌های تقسیم سلول را تعیین کند و آن را از یک فایل بخواند.
- پیاده‌سازی پروژه در فایل‌های جداگانه، در این روش توابع مرتبط با یکدیگر در یک فایل h. تعریف شده و در فایل c. متناظر پیاده‌سازی می‌شوند.
- ذخیره و خواندن نقشه از روی یک فایل text که برای انسان‌ها نیز قابل خواندن باشد. (فرمت فایل به عهده شماست.) در این حالت موظف هستید یک برنامه جداگانه جهت تبدیل نقشه باینری داده شده به نقشه خودتان توسعه دهید.
- پیاده‌سازی زیباتر نقشه و نمایش آن به صورت گرافیکی نمره‌ی اضافی خواهد داشت. در این قسمت دست شما برای استفاده از هر کتابخانه‌ای در زبان C باز است.
- هر ایده‌ی دیگری که توسط خودتان به پروژه اضافه شود با توجه به حجم آن نمره‌ی اضافی خواهد داشت.
- پیاده‌سازی حالت بازی با کامپیوتر، در این حالت کامپیوتر به جای کاربر دوم تصمیم می‌گیرد.
- پیاده‌سازی عملیات undo به تعداد مشخص یا نامشخص. در این حالت کاربر می‌تواند یکی از تصمیمات خود را undo کند. در نظر داشته باشید که این کار تنها برای بازی انفرادی ممکن است.
- استفاده از Git در صورت داشتن تعداد منطقی Commit با پیام‌های مشخص و مرتبط با پروژه نمره‌ی اضافه خواهد داشت. بهتر است یکی از تدریس‌یاران را به پروژه خودتان اضافه کنید تا ایشان روند پروژه را نظارت کنند.

## قدردانی

نسخه‌ی اولیه این پروژه به قلم ایمان تبریزیان و شیوا زمانی بوده است که جا دارد اینجا از آن‌ها تشکر شود.

آرزوی موفقیت روزافزون در تمامی لحظات پیشرو

تیم تدریسیاری

پاییز ۱۳۹۸