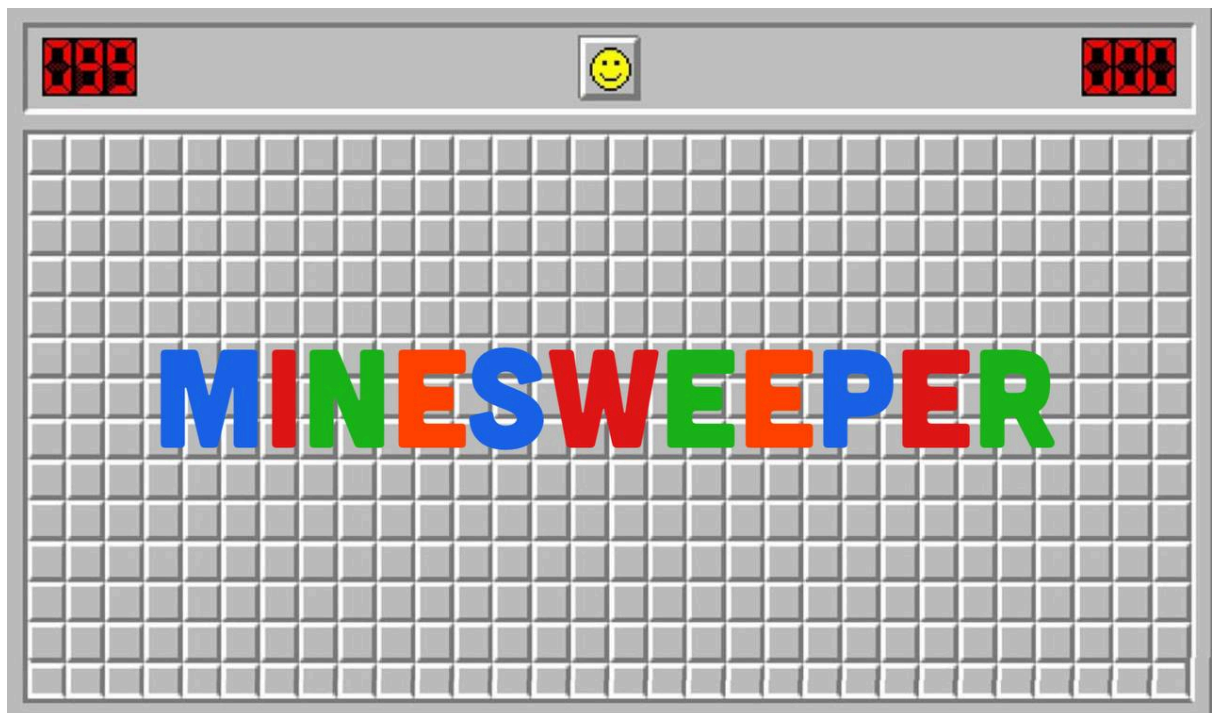




Basics of Programming Mini Project Winter 2025

Course Instructor: Dr. Saeed Reza Kheradpisheh



سلام! امیدواریم حالتون خوب باشه و تا اینجا ترم خوب و مفیدی رو گذرونده باشید.

در طول ترم مباحث مختلفی رو در برنامه نویسی یاد گرفتید و حالا قراره با انجام یک مینی پروژه که به صورت تک نفره هست بیشتر با چیزایی که یاد گرفتید کار کنید و خودتون رو به چالش بکشید تا آماده پروژه اصلی بشید.

توجه کنید که انجام این پروژه به شما در پیاده سازی پروژه پایانی بسیار کمک خواهد کرد، پس تا حد امکان برایش وقت بگذارید.

خب بریم سراغ اصل مطلب:

1. مقدمه و آشنایی با بازی

همونطور که از صفحه اول حدس زدید، بازی Minesweeper هست اما برای کسانی که ممکنه با بازی آشنایی نداشته باشند و یا یادشون رفته باشه، تمام قوانین و نحوه بازی در ادامه به صورت دقیق توضیح داده میشه:

Minesweeper یک ژانر بازی ویدیویی معمایی منطقی شامل شبکه‌ای از کاشی‌های قابل کلیک است که "مین"‌های پنهان در سراسر صفحه آن پخش شده‌اند. هدف این است که بدون منفجر کردن هیچ مینی، صفحه بازی را پاکسازی کنید، در حالی که سرخ‌هایی درباره تعداد مین‌های مجاور در هر خانه به شما کمک می‌کنند.

● شروع بازی:

بازی با یک شبکه مستطیلی از بلوک‌های مربعی شروع می‌شود. بازیکن باید با کلیک کردن روی یکی از بلوک‌ها بازی را آغاز کند. اولین کلیک همیشه امن است و یک فضای خالی یا عددی را نشان می‌دهد که تعداد مین‌های اطراف آن بلوک را مشخص می‌کند.

دقت کنید که همیشه باید حداقل یک محدوده دور بلوکی که کاربر روی آن کلیک کرده امن باشد.

● انواع بلوک‌ها:

1. بلوک خالی (یا صفر): نشان‌دهنده فضای است که هیچ مین در اطراف آن وجود ندارد.

2. بلوک عددی: عددی که نشان می‌دهد چند مین در مجاورت آن بلوک قرار دارند (اعداد بین 1 تا 8).

3. مین (Mine): اگر روی بلوکی که مین است کلیک کنید، بازی به پایان می‌رسد.

4. بلوک پرچمدار: بلوکی که بازیکن آن را به عنوان مشکوک به داشتن مین علامت‌گذاری کرده است.

● نحوه تغییر بلوک‌ها:

باز کردن: بلوک انتخابی را باز می‌کند. اگر بلوک عددی باشد، عدد مشخص می‌شود. اگر مین باشد، بازی تمام می‌شود.

پرچمدار کردن: بلوک باز نشده را با پرچم علامت‌گذاری می‌کند. این کار برای شناسایی بلوک‌هایی که احتمالاً مین دارند استفاده می‌شود. و یا بلوک پرچمدار شده را به حالت اولیه باز می‌گرداند.

- هدف بازی:
هدف این است که تمام بلوک‌هایی که مین ندارند را باز کنید، بدون اینکه روی بلوکی با مین کلیک کنید. بازیکن باید با استفاده از اعداد و منطق، جایگاه دقیق مین‌ها را شناسایی کند و بلوک‌های دیگر را باز کند.
- باخت در بازی:
اگر بازیکن روی یک بلوک حاوی مین کلیک کند، بازی تمام می‌شود و تمام مین‌های باقی‌مانده در شبکه نمایش داده می‌شوند.

قبل این که ادامه بدید به شدت پیشنهاد میشه بازی رو توی اینترنت تست کنید تا باهاش بیشتر آشنا بشید، حواستون باشه که درک بازی ای که دارید می سازید اهمیت ویژه ای دارد و در روند پروژه کمکتون می‌کنه و می‌تونه بهتون ایده‌های جالبی بده.

می‌تونید از لینک‌های زیر به بازی دسترسی داشته باشید:

[سایت](#) | [اندروید](#) | [iOS](#)

2 - توضیحات پیاده سازی

چون این پروژه اولین پروژه شماست سعی می‌کنیم به شما یک تقسیم بندی پیشنهادی ارائه بدیم ولی شما آزاد هستید خودتون طبق برنامه ریزی ای که دوست دارید و یا با منتور تون هماهنگ می‌کنید پروژه رو پیش ببرید.

فاز 1: منو (Menu)

ابتدا باید یک منو ایجاد کنید که شامل گزینه‌هایی مانند “Leaderboard”، “New Game” و “Exit” باشد. همچنین باید توضیحی برای کاربر قرار دهید تا چگونگی استفاده از منو و بازی شما را بدانند. برای مثال، می‌توانید یک راهنمای کوتاه بنویسید که توضیح دهد هر گزینه چه کاری انجام می‌دهد.

1- New Game

بعد زدن شروع بازی، باید اسم کاربر پرسیده شود تا بعدا به همراه امتیاز کاربر در جدول امتیازات ثبت شود. سپس کاربر وارد محیط بازی جدید می‌شود.

2- Leaderboard

در این بخش باید تمام بازی‌ها را به همراه نام کاربر و امتیازش هنگام پایان بازی رنک بندی کنید. یعنی از بیشترین امتیاز تا کمترین امتیاز به ترتیب مرتب کنید و نمایش دهید.

نوع لیدربرد و نمایش امتیاز شما می‌تواند با توجه به ویژگی های امتیازی متفاوت و پیشرفته‌تر باشد.

3- Exit

با انتخاب این گزینه برنامه در ترمینال بسته می‌شود.

فاز 2: محیط بازی

محیط بازی باید شامل موارد زیر باشد:

1. شبکه‌ای از خانه‌ها:

- یک جدول که خانه‌های آن نمایانگر سلول‌های بازی هستند.
- سلول‌ها در ابتدای بازی باید پنهان باشند و فقط با انتخاب کردن هر کدام، محتوای آن‌ها (مین یا عدد) نشان داده شود.

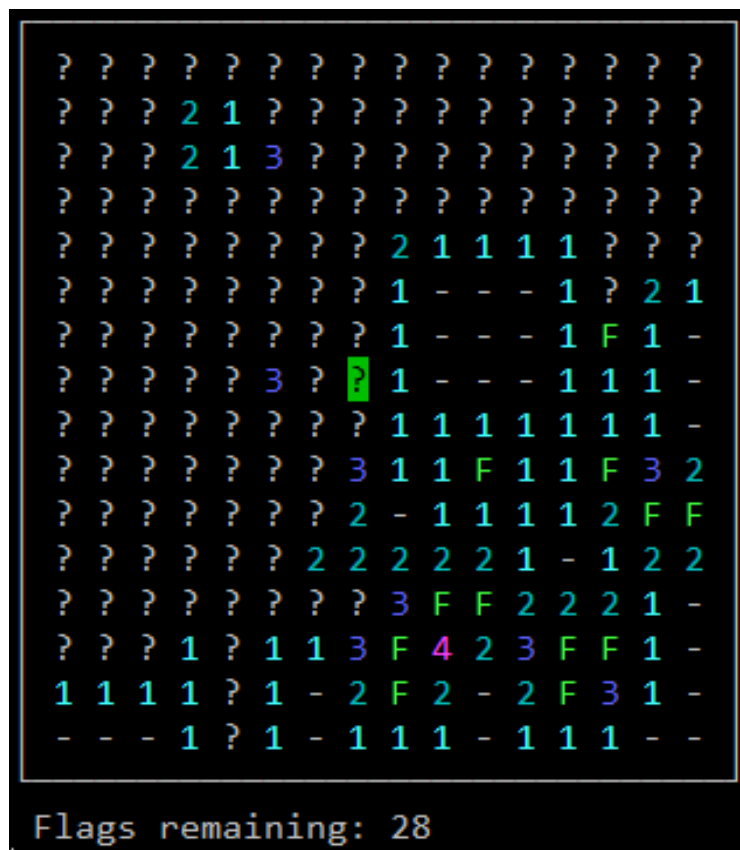
2. نمادها:

برای هر یک از حالت‌های زیر، از نماد خاصی برای نشان دادن آن استفاده کنید:

- خانه‌های پنهان که هنوز نمایش داده نشده‌اند
- خانه‌های خالی (خانه‌هایی که هیچ مینی در مجاورت خود ندارند)
- خانه‌های شمار مدار با اعداد ۱ تا ۸ (تعداد مین‌های اطراف آن سلول)
- مین (تنها زمانی که کاربر بازی را باخته باشد)

نمونه‌ای از محیط بازی در ترمینال به صورت زیر است.

(ظاهر بازی شما می‌تواند کاملاً متفاوت باشد، این تصویر را فقط به عنوان یک مثال در نظر بگیرید!)



فاز 3: جایگذاری مین‌ها

بعد از ساخت جدولی مستطیلی شکل به عنوان صفحه بازی، تعدادی از خانه و یا سلول‌ها باید به صورت رندوم انتخاب شده و به عنوان خانه‌های حاوی بمب علامت گذاری شوند. دقت کنید که بازی بعد از اولین کلیک بازیکن شروع شده و اولین خانه انتخاب شده همیشه خانه‌ای خالی و یا دارای عدد می‌باشد بنابراین پروسه انتخاب تصادفی خانه‌ها برای درج بمب باید بعد از شروع بازی و اولین کلیک بازیکن انجام شود.

فاز 4: نشان گذاری خانه ها و چاپ اعداد

هر خانه‌ای که شامل مین نیست، باید یا خالی باشد و یا دارای عددی بین 1 تا 8، که نشانگر تعداد مین‌های موجود در خانه‌های اطراف آن می‌باشد. توجه داشته باشید برای انتخاب خانه های جدول بازی در ترمینال با توجه به نبود کاربرد ماوس باید از دکمه های کیبورد برای جا به جایی روی صفحه استفاده کنید.

زمانی که کاربر یک سلول را انتخاب می‌کند:

- اگر سلول مین باشد، تمام مین‌ها منفجر می‌شوند و بازی تمام می‌شود.
- اگر سلول شامل عدد باشد، تنها همان عدد که نشان دهنده تعداد بمب‌ها در مجاورت ضلعی و یا مجاورت قطری آن سلول است نمایش داده شود.
- اگر سلول خالی باشد (هیچ مینی در مجاورت آن سلول نباشد)، خانه‌های مجاور از هر جهت شروع به نمایان شدن می‌کنند، و بعد از نمایان شدن اولین خانه‌ی شماردار، گسترش در آن جهت متوقف می‌شود.

فاز 5: شرایط برد و باخت

برد بازیکن در صورتی رخ می‌دهد که تمام سلول‌های خالی و عدد دار جدول بازی انتخاب و باز شده و در حال نمایش باشند. حتی اگر تمام سلول‌های بمب‌دار پرچمدار شده باشند اما یک سلول خالی و یا عدد دار هم موجود باشد که توسط بازیکن باز نشده باشد، بازی به اتمام نرسیده و ادامه دارد.

باخت بازیکن نیز هنگامی رخ می‌دهد که بازیکن یکی از سلول‌های بمب‌دار را انتخاب و باز کند.

در صورت باخت و یا برد بازیکن، بازی به پایان می‌رسد.

فاز 6: محاسبه امتیاز

امتیازدهی در این بازی بر اساس مدت زمان حل بازی است. هرچه بازیکن سریع‌تر تمام مین‌ها را شناسایی کند، امتیاز بالاتری کسب می‌کند. جزئیات به شرح زیر است:

1. شروع زمان: تایمر بلافاصله پس از کلیک اولین خانه توسط بازیکن شروع می‌شود.
2. پایان زمان: تایمر زمانی متوقف می‌شود که بازیکن همه سلول های امن را باز کرده باشد و برنده شده باشد.
3. فرمول امتیاز: امتیاز به صورت نسبت عکسی از زمان محاسبه می‌شود. زمان‌های سریع‌تر منجر به امتیازات بالاتری می‌شوند.

در پایان بازی، امتیاز کاربر به همراه نام او در جدول امتیازات ذخیره می‌شود.

دقت کنید که در صورت باخت بازیکن، اطلاعات او یعنی نام و طول زمان بازی و اینکه بازیکن نتوانسته بازی را به اتمام برساند باید در جدول امتیازات مشخص شود.

فاز 7: جدول امتیازات

جدول امتیازات و یا همان leaderboard بخشی از برنامه شماست که در آن هر بازی به همراه اسم بازیکن، برد یا باخت او و در صورت برد، امتیاز او و مدت زمان بازی‌اش به ترتیب امتیاز ذخیره و نمایش داده می‌شود. واضح است که بازنده‌ها در پایان لیست قرار می‌گیرند. برای ذخیره کردن اطلاعات هر بازی شما می‌توانید از دو روش استفاده کنید.

1. در حافظه برنامه در حال اجرا، هر بازی و اطلاعات آن را در متغیرهایی ذخیره کنید و در بخش جدول امتیازات آن‌ها را نمایش دهید. واضح است که در این صورت با پایان و خروج از برنامه اطلاعات بازی‌های ذخیره شده از بین خواهد رفت و با اجرای دوباره برنامه، بازیکن‌های جدید توانایی مشاهده امتیازات بازیکن‌های قبلی را نخواهند داشت.
 2. روش پایدارتر و بهتر این است که هر بازی و اطلاعات آن را در یک **فایل** ذخیره کنید. و با هر بار شروع مجدد و اجرا کردن برنامه، اطلاعات فایل را بخوانید و جدول امتیازات را بازنویسی کنید تا امتیازات تمام بازی‌ها قابل مشاهده باشد. البته که این مورد امتیازی است اما شدیداً پیشنهاد می‌شود که از این روش استفاده کنید. (این مورد در پروژه پایانی شما اجباری خواهد بود پس بهتره از همین الان برای کار با فایل تمرین کنید)
- جدول امتیازات باید با هر بازی جدید انجام شده بروزرسانی شود تا درست‌ترین رده‌بندی بازی‌ها را نمایش دهد.

3. نکات

- سعی کنید تا حد امکان کد تمیز و مرتبی بنویسید. (قواعد **کلین کد** را رعایت کنید).
- کد تکراری ننویسید (DRY : Don't Repeat Yourself) و از توابع برای رفع این مشکل استفاده کنید.
- بعد از نوشتن کد هر بخش و قبل از پیاده سازی بخش دیگر، حتماً بخشی که نوشتید رو تست کنید و همه رو آخر کار نذارید تا دیباگ کردن و ادامه کارتون راحت‌تر باشد.
- برای ذخیره کردن اطلاعات بازی حتماً باید از **استراکت** استفاده کنید.
- برای گرفتن ورودی تنها یک کاراکتر از کیبورد، بدون صبر کردن برای فشردن دکمه‌ی Enter باید از تابع `getch` استفاده کنید که راهنمایی استفاده از این تابع رو می‌تونید در بخش توابع و کتابخانه ببینید.
- برای سیستم عامل ویندوز شما با استفاده از دستور `system("cls")` می‌تونید تمام چیزهایی که تو ترمینال نوشتید رو پاک کنید. برای سیستم عامل‌های مک و لینوکس هم با استفاده از دستور `system("clear")` می‌تونید تمام چیزهایی که تو ترمینال نوشتید رو پاک کنید.

4. توابع و کتابخانه‌های کاربردی

- **تابع random:** از کتابخانه `<cstdlib>` استفاده می‌کند و به منظور تولید اعداد تصادفی استفاده می‌شود. این تابع یک عدد تصادفی در بازه مشخص را تولید می‌کند. برای استفاده از تابع `random`، ابتدا باید کتابخانه مربوطه را به کد خود اضافه کنید:

```
#include <cstdlib>
#include <ctime>
```

سپس در ابتدای برنامه یا تابع main، باید seed تابع را مقداردهی اولیه کنید:

```
srand(static_cast<unsigned int>(time(nullptr)));
```

می‌توانید تابع rand را برای تولید اعداد تصادفی استفاده کنید. مثلاً برای تولید عدد تصادفی بین 1 تا 100:

```
int randomNum = rand() % 100 + 1;
```

- **تابع getch:** برای دریافت یک کاراکتر از ورودی بدون نمایش آن در کنسول استفاده می‌شود. برای استفاده از تابع getch، ابتدا باید کتابخانه مربوطه را به کد خود اضافه کنید:

```
#include <conio.h>
```

می‌توانید از تابع getch برای دریافت یک کاراکتر بدون نمایش آن استفاده کنید:

```
char userInput = getch();
```

همچنین به این نکته هم توجه کنید که حتماً باید از کلیدهای WASD برای حرکات بازی استفاده کنید.

- **کاراکترهای Unicode:** برای اینکه بتوانید طراحی قشنگی در ترمینال نمایش دهید باید از کاراکترهای box-drawing (لینک) و همچنین دیگر کاراکترهای یونیکد (لینک) استفاده کنید.

	0	1	2	3	4	5	6	7	8	9	A	B	C	D
U+250x	—	—			---	---			---	---			┌	┐
U+251x	┐	┐	┐	┐	└	└	└	└	└	└	└	└	└	└
U+252x	└	└	└	└	┐	┐	┐	┐	┐	┐	┐	┐	┐	┐
U+253x	┐	┐	┐	┐	└	└	└	└	└	└	└	└	└	└
U+254x	┐	┐	┐	┐	┐	┐	┐	┐	┐	┐	┐	┐	--	--
U+255x	=		└	└	└	└	└	└	└	└	└	└	└	└
U+256x	└	└	└	└	└	└	└	└	└	└	└	└	└	└
U+257x	└	└	└	└	└	└	└	└	└	└	└	└	└	└

راهنمایی: برای چاپ کردن این کاراکترها پس از پیدا کردن کدشان باید به اینصورت عمل کنید:

```
cout << "\u{UNICODE}";
```

for example:

```
cout << "\u2551";
```

برای این که ترمینال "ویندوز" کاراکترهای box-drawing را بفهمد و درست چاپ کند، لازم است که این کارها را در سیستم‌تان انجام دهید:
کنترل پنل را باز کنید و به تنظیمات Region بروید.

از قسمت administrative و از بخش language for non-unicode programs گزینه system local change را انتخاب کنید و تیک use Unicode UTF-8 را بزنید.
یک راه جایگزین و بهتر برای اینکه مستقل از نوع سیستم که برنامه روی آن در حال اجرا شدن است، این هست که خط زیر را اول برنامه خود در واقع خط اول main بنویسید:

```
SetConsoleOutputCP(CP_UTF8);
```

این تابع و بسیاری دیگر از توابعی که داخل پروژه ها استفاده خواهید کرد داخل کتابخانه windows.h خواهند بود و بخشی از کاربری های سیستم عامل ویندوز است. اگر از سیستم عامل دیگری مثل لینوکس یا مک استفاده می کنید پیشنهاد ما به شما استفاده از یک virtual machine و یا استفاده از کتابخانه های دیگر مثل conio.h است هر چند راه اول قطعاً ساده تر خواهد بود.

5. موارد امتیازی

1. استفاده از فایل برای ذخیره سازی جدول امتیازات مهم ترین مورد امتیازی این تمرین است.
2. زیبایی بصری بازی و کاربر پسند بودن آن نمره امتیازی دارد؛ البته، مشخصاً باید یک سری از حداقل ها را رعایت کنید و بعد از آن، نمره امتیازی می شود.
3. به جای اینکه سائز صفحه بازی یک عدد ثابت باشد، می توانید با گرفتن دو عدد از کاربر مانند n و m ، سائز صفحه را به صورت $n * m$ نمایش دهید و بازی در این صفحه انجام بگیرد. در این صورت به تعداد مین های داخل بازی نسبت به سائز صفحه توجه کنید.
4. هنگام ذخیره کردن اطلاعات بازیکن ها، برای تمایز بین بازی های ثبت شده می توانید تاریخ و ساعت انجام هر بازی را نیز در کنارش نمایش دهید.
5. درجه سختی: می توانید برای بازی درجه های سختی متفاوتی تعیین کنید که کاربر توانایی انتخاب آن را داشته باشد.
6. می توانید به جای تعداد مین های اطراف هر بلوک، مین های آن سطر یا ستون را نشان دهید تا سطح متفاوتی از سختی را پیاده سازی کرده باشید.
7. نمایش تعداد مین های باقی مانده و یا هینت های دیگر
8. پیاده سازی یک سیستم برای خرید هینت، کمک و یا reset کردن یک حرکت بر اساس پرداخت سکه که سکه ها را می توان از برد در بازی به دست آورد.
9. محدودیت زمانی برای پیدا کردن تمامی مین ها
10. محدودیت زمانی برای هر حرکت بازیکن
11. درجه سختی دینامیک: درجه سختی در یک بازی می تواند در گذر زمان به طور تدریجی افزایش یابد، پیاده سازی این سطوح مختلف از سختی بازی وابسته به الگوریتم طراحی شده و خلاقیت خود شما دارد.
12. دونفره بودن بازی: به این صورت که دو بازیکن نوبتی بر روی یک صفحه بازی می کنند، و هر کس محدودیت زمانی دارد؛ در نهایت فردی که بیشترین مین ها را پیدا کند برنده است (افراد می توانند از نشانه گذاری اشتباه برای گیج کردن حریف استفاده کنند). جدول امتیازات بازی های دو نفره باید از جدول های دیگر جدا باشد. همچنین توانایی جا به جایی همزمان روی صفحه از طریق مجموعه کلید های مختلف کیبورد ممکن باشد.

13. سیستم امتیازدهی پیشرفته‌تر: علاوه بر زمان، عوامل دیگری از جمله تعداد حرکت‌های بازیکن و سطح سختی بازی در تعیین امتیاز بازیکن نقش داشته باشند.

14. قابلیت نمایش جدول امتیازات به صورت دلخواه، مرتب شده بر اساس بالاترین امتیاز تا پایین‌ترین، جدیدترین بازی تا قدیمی‌ترین، حروف الفبا اسم بازیکن‌ها و ...

15. در نهایت هر ایده جدیدی که خودتون دارید و به ذهنتون میرسه رو می‌تونید پیاده سازی کنید و هنگام ارائه، اون رو توضیح بدید و نمره‌ش رو بگیرید.

دقت کنید که اگر مورد امتیازی پیاده سازی کردید باید حتما به منتورتون اطلاع بدید.

6. ارزیابی

1. کلین کد و خوانا بودن و در کل کیفیت کد بخش مهمی از نمره شما را تشکیل می‌دهد. پس تا حد امکان سعی کنید به بهترین روش ایده هاتون رو پیاده سازی کنید!

2. پیاده سازی نکاتی که در فازبندی گفته شده و همینطور درست اجرا شدن بازی و پیاده سازی درست منطق این بازی هم بخش مهم دیگری از نمره این پروژه خواهد بود. فقط در صورتی سراغ بخش امتیازی بروید که بخش اجباری پروژه را انجام داده باشید.

3. استفاده درست از استراکت (و فایل در صورتی که ازش استفاده کردید) نمره‌ی ویژه خودش را دارد.

4. نمره دهی هر فرد **توسط منتور خودش** انجام خواهد شد.

سعی کنید مدیریت خوبی روی پروژه داشته باشید و این رو تمرین کنید که برای ادامه کارتون به مشکل نخورید. در آخر پروژه می‌فهمید که کد زدن بدون مدیریت درست پروژه، به نتیجه خوبی نمی‌رسه و همونقدر که برنامه نویسی خوب مهمه، مدیریت خوب پروژه هم مهمه! پیشنهاد ما اینه که مدیریت پروژه رو خودتون انجام بدید و صرفا برنامه‌هاتون رو برای منتورتون توضیح بدید که کمکتون کنه تا مدیریت رو کم کم و با تجربه به دست بیاورید پس چه بهتر که از همین الان شروع به تمرینش کنید!

دولاین پروژه تا **19 دی** هست و باید کدهاتون رو به موقع در کوئرا آپلود کنید.

موفق باشید!