



بسمه تعالی



مستند مسابقه

دهمین نبرد هوش مصنوعی شریف

مرحله ی غیر حضوری

بهمن و اسفند ۱۳۹۶





## سرزمین سه خورشید

هزاران سال آینده در سیاره‌ی دورافتاده‌ی تریسول واقع در کنج ممنوعه‌ی کهکشان، مردمانی از جنس آب در محیطی گرم و کویری زندگی می‌کنند. گرمای تریسول به این دلیل است که سیاره حول سه ستاره می‌چرخد. هر یک از این خورشیدها یک چرخه‌ی روز و شب متفاوت دارند و در مدت زمانی که تقریباً برابر ۷ روز زمینی است، هر سه خورشید پشت سیاره قرار می‌گیرند و شب تاریک رخ می‌دهد. در شب تاریک نوری درخشان از بدن مایع مردمان سیاره ساطع می‌شود و زندگی در تاریکی را ممکن می‌سازد. مردمان ساکن تریسول از چشمه‌ی واقع در شمال سیاره متولد می‌شوند. هر بار که سال نو فرامی‌رسد، چشمه‌ی حیات می‌جوشد و جویباری را از کوه به دریاچه‌ی حیات جاری می‌کند. تریسولیان تازه متولد از دریاچه‌ی حیات برمی‌خیزند و در عرض چند ماه به بلوغ کامل می‌رسند.



ساختار اجتماعی تریسولیان ساده است و امور روزمره طبق دستورات سلطان پیش می‌رود. سنت دیرین مردمان سیاره حکم می‌کند که هر کس سلطان را به قتل برساند، می‌تواند به جای او بر تخت سلطنت تکیه کند. به همین دلیل افرادی که به قدرت می‌رسند، در زمان کمی کشته می‌شوند و طول دوره‌ی هر حکومت به طور میانگین چند روز است. صدراعظم و کشیش اول به سلطان، که معمولاً از تجربه‌ی کافی برخوردار نیست، در اداره‌ی مسائل داخلی و خارجی سیاره کمک می‌کنند. کشیش اول و صدر اعظم فعلی، که

مدت زیادی ست این مناصب را بر عهده دارند، به ترتیب رفتاری جاه‌طلبانه و محتاطانه دارند و همواره سعی می‌کنند که نظر سلطان ساده‌لوح را به نظر خود نزدیک کنند.

داستان از اینجا شروع می‌شود که کشیش اول سرنخ‌هایی از دلیل درخشش بدن تریسولیان در حین شب تاریک به دست می‌آورد. وی متوجه می‌شود که دنیایی موازی با دنیایی که می‌شناسند وجود دارد و اجزای دنیای موازی در عین اینکه از لحاظ مکانی و زمانی به دنیای معمولی تنیده شده‌اند، نسبت به اجزای این دنیا تا حدی متفاوت هستند. کشیش کشف می‌کند که در دنیای موازی مردمانی از جنس آتش در سیاره‌ای سرد و مرطوب زندگی می‌کنند، این سیاره و مردمانش از هر نظر به جز جنس و ماهیت، مشابه دنیای معمولی هستند. شهر دنیای سرد که



کشیش آن را گلو ترا می‌نامد، مشابه شهر دنیای کویری است و به‌ازای هر فردی که در یکی از این دنیاها وجود دارد، فردی در



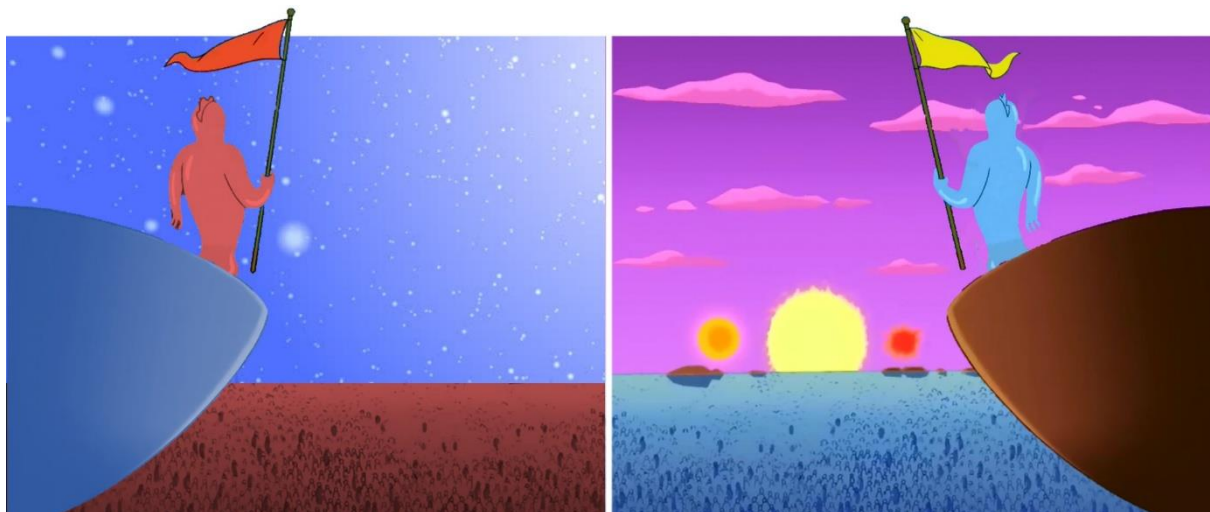


دنیای دیگر نیز زندگی می‌کند. کشیش پی می‌برد که در شب تاریک به دلیل سرمای هوا در دنیای کویری، این دو دنیا در نزدیک‌ترین حالت به یکدیگر قرار می‌گیرند و به همین علت مردمان تریسول، ماهیت موازی خودشان که از جنس آتش است را در درون احساس می‌کنند و این مسئله باعث به وجود آمدن نور گرم در بدن آن‌ها می‌شود.

کشیش که از کشف این مسئله به وجد آمده است، سریعاً خود را به سلطان می‌رساند. پیشنهاد او این است که سلطان نیروها را برای حمله به دنیای سرد آماده کند. کشیش با اشتیاق پیشنهاد خود را توجیه می‌کند و می‌گوید که محیط سرد و مرطوب جهان موازی برای زندگی مردم ایده‌آل است و دیگر هیچ‌گاه نباید نگران تبخیر شدن از گرمای شدید سه خورشید باشند. صدراعظم که این مکالمه را می‌شنود، اعتراض می‌کند و می‌گوید که نیروهای سلطنتی نباید درگیر یک دنیای خیالی شوند تا توان دفاع از شهر، مردمان و چشمه‌ی حیات را از دست دهیم. سلطان در نهایت تصمیم می‌گیرد که بودجه را مشترکاً برای دفاع به صدراعظم و برای حمله به کشیش تخصیص دهد، سپس به کشیش دستور می‌دهد که کار روی درگاه به دنیای موازی برای حمله را شروع کند و به صدراعظم دستور می‌دهد که روی ساختارهای دفاعی شهر کار کند.

کشیش بعد از چند ماه کار در حاشیه‌ی شهر، موفق می‌شود که دروازه‌ای به دنیای موازی باز کند. نور کورکننده‌ای به وجود می‌آید و تصویری از دنیای موازی برای لحظه‌ای نمایان می‌شود. در کسری از ثانیه کشیش در عین ناباوری فردی را در آن سو می‌بیند که او نیز از دیدن کشیش حیرت‌زده است. سرمای دنیای موازی تمام وجود کشیش را فرا می‌گیرد و واقعیت ترسناک بر او آشکار می‌شود. فردی در دنیای موازی نیز به طور هم‌زمان و با انگیزه‌ی مشابه، دروازه‌ای به این دنیا گشوده است و قصد حمله به سیاره‌ی تریسول را دارد ...

حال که جنگی تمام عیار شکل گرفته و دو دنیا قصد تسخیر یکدیگر را دارند، شما وظیفه دارید تا در نقش فرمانروایان تریسول یا گلو ترا به جنگ با یکدیگر پردازید و در عین تاختن به دنیای حریف از دنیای خود دفاع کنید.





## وضعیت بازی:

در هر لحظه، دو نقشه‌ی مشابه به طور موازی وجود دارد. هر بازیکن در یکی از نقشه‌ها به دفاع و در دیگری به حمله می‌پردازد و تنها نقطه‌ی ارتباط بین این دو فرآیند این است که بودجه‌ی بازیکن برای حمله و دفاع مشترک است. پس برای سادگی کار در ادامه‌ی توضیحات بازی تنها به اتفاقات یک نقشه اشاره می‌کنیم.

پارامتر  $x$ ، عرض نقشه و پارامتر  $y$ ، ارتفاع نقشه را مشخص می‌کند. خانه مبدا  $(0,0)$  نیز در بالا و چپ نقشه قرار دارد.

$x$

در هر لحظه از بازی، اجزای زیر در نقشه وجود دارند:

**خانه‌های هموار:** نقاطی از نقشه هستند که مدافع می‌تواند در آن‌ها برج بسازد.

**خانه‌های مسدود:** نقاطی از نقشه هستند که با وجود اینکه جاده‌ای از آنان نمی‌گذرد، امکان ساختن برج در آن‌ها وجود ندارد.

$y$

**جاده‌ها:** دنباله‌ای پیوسته از خانه‌های نقشه هستند که خانه ابتدایی و انتهایی متمایز دارند. طبیعتاً امکان ساختن برج در مسیر جاده‌ها وجود ندارد. جاده‌های یک نقشه می‌توانند یک‌دیگر را قطع کنند و حتی در قسمتی از مسیر با یک‌دیگر اشتراک داشته باشند.

**برج‌ها:** عوامل دفاعی بازیکن مدافع هستند که در خانه‌های هموار ساخته شده‌اند. دقت کنید که در هر خانه حداکثر یک برج می‌تواند احداث شود. با احداث هر برج، دیگر نمی‌توان در خانه‌های مجاور (ضلعی) برج دیگری ساخت و این خانه‌ها به خانه مسدود تبدیل می‌شوند. برج‌ها از دو نوع «توپ جنگی» و «کمان‌دار» هستند و دارای یک عدد «تراز» هستند که روی توانایی‌های آن‌ها تاثیر می‌گذارد.

**سربازان:** درون هر خانه از هر مسیر تعدادی سرباز می‌تواند قرار بگیرد. سربازها از دو نوع «زره‌پوش» و «عادی» هستند و مشابه برج‌ها دارای یک تراز می‌باشند.

در هر نقشه بازیکنی که به دفاع می‌پردازد، می‌تواند وضعیت نقشه را تماماً مشاهده کند. اما بازیکن مهاجم در عین اینکه بقیه‌ی اجزا را کاملاً مشاهده می‌کند، اطلاعات ناقصی از برج‌های مدافع در اختیار دارد و برای دیدن یک برج باید حداقل یک بار، یکی از سربازان خود را تا نزدیکی آن برده باشد.





## روند درگیری:

وضعیت توصیف شده در بالا را در نظر بگیرید. می‌دانیم که هر سرباز با تراز مشخص شده، ویژگی‌های زیر را دارد:

### ویژگی سربازها:

#### سلامتی اولیه:

یک سرباز در ابتدا از این میزان سلامتی برخوردار است.

#### سلامتی فعلی:

یک سرباز در حال حاضر از این میزان سلامتی برخوردار است.

#### دوره‌ی حرکت:

بعد از هر قدم به سمت جلو، سرباز باید به این میزان واحد زمانی صبر کند تا قدم بعدی را بردارد.

#### جایزه‌ی کشتن:

میزان پولی که در صورت کشته شدن این سرباز، به تیم مدافع داده می‌شود.

#### پتانسیل آسیب:

آسیبی که سرباز در صورت به اتمام رساندن مسیر به تیم مدافع وارد می‌کند.

#### محدوده دید:

اگر برجی در این فاصله از سرباز قرار گیرد به برج‌های قابل مشاهده توسط تیم مهاجم اضافه می‌شود.

و می‌دانیم که هر برج با تراز مشخص شده ویژگی‌های زیر را دارد:

### ویژگی برج‌ها:

#### قدرت تخریب:

هرگاه یک برج یک سرباز را هدف قرار دهد، به میزان قدرت تخریب برج، سلامتی سرباز کاهش می‌یابد.

#### دوره شلیک:

یک برج بعد از هر شلیک، باید به میزان مشخصی از واحد زمانی صبر کند تا بتواند شلیک بعدی را انجام دهد.





**تیررس:**

اگر فاصله‌ی یک خانه تا برج، از تیررس آن کمتر یا مساوی باشد، به این معناست که برج می‌تواند سربازان مستقر در آن خانه را هدف قرار دهد. معیار فاصله جمع اختلاف افقی و عمودی یا به عبارتی فاصله‌ی منهتنی است.

**هدف کنونی:**

در هر لحظه از بازی، یک برج ممکن است سربازی را به عنوان هدف فعلی داشته باشد.

با توجه به ویژگی‌های توصیف شده برای برج‌ها و سربازان، درگیری در هر واحد زمانی به این شکل پیش می‌رود:

هر واحد زمانی را یک نوبت می‌نامیم، در هر نوبت ابتدا برج‌ها و سپس سربازان فعالیت می‌کنند.

**فعالیت سربازها:**

هر سرباز در انتهای هر دوره‌ی حرکت خود، در مسیری که روی آن قرار دارد یک واحد به سمت جلو حرکت می‌کند و در غیر این صورت در مکان فعلی خود باقی می‌ماند. در هنگام ساخت، سرباز از نظر زمانی در ابتدای دوره‌ی حرکت خود قرار دارد. اگر بعد از حرکت رو به جلو، برجی در فاصله‌ی قدرت دید سرباز قرار بگیرد، به برج‌های قابل مشاهده توسط تیم مهاجم اضافه می‌شود. همچنین اگر سربازی روی خانه‌ی آخر مسیرش مستقر باشد و اقدام به حرکت کند، از نقشه خارج شده و به میزان پتانسیل آسیبی که دارد به تیم مدافع ضربه می‌زند.

**فعالیت برج‌ها:**

تفاوت اصلی برج کمان‌دار و توپ جنگی این است که اگر توپ جنگی سربازی را مورد هدف قرار دهد، تمام سربازانی که هم‌خانه‌ی سرباز مذکور هستند نیز به طور خودکار مورد هدف قرار می‌گیرند. این در حالی‌ست که کمان‌دار فقط به یک تک سرباز شلیک می‌کند. هر برج اگر از دوره‌ی شلیکش باقی مانده باشد کاری انجام نمی‌دهد. در غیر این صورت اگر هدف کنونی داشته باشد و آن هدف هنوز در تیررس برج باشد، به آن شلیک می‌کند. اگر هدف کنونی نداشته باشد یا از تیررس خارج شده باشد، یک هدف کنونی جدید انتخاب می‌کند و به آن شلیک می‌کند. اولویت معیارهای مختلف برای انتخاب هدف به ترتیب زیر است. دقت کنید که شرط اصلی انتخاب هدف در تیررس قرار داشتن آن است و معیارهای زیر برای مقایسه‌ی اهداف در تیررس هستند. طبیعتاً اولویت  $k$  ام زمانی ارزش پیدا می‌کند که دو هدف مورد مقایسه در  $k-1$  اولویت قبلی یکسان باشند.

**اولویت اول:**

همانطور که گفته شد، اگر هدف کنونی برج در تیررس باشد حتماً انتخاب می‌شود.

**اولویت دوم:**

سربازی که به انتهای مسیر خود نزدیک‌تر است، برای مورد هدف قرار گرفتن اولویت بیشتری دارد.





اولویت‌های بعدی برای دو نوع برج توپ جنگی و کمان‌دار متفاوت است:

#### اولویت سوم کمان‌دار:

سرباز زره‌پوش نسبت به سرباز عادی اولویت دارد.

#### اولویت چهارم کمان‌دار:

سربازی که سلامتی کمتری دارد نسبت به سرباز سالم‌تر اولویت دارد.

#### اولویت سوم توپ جنگی:

اگر یک سرباز در خانه‌ای قرار داشته باشد که تعداد بیشتری سرباز را در خود گنجانده باشد، به سرباز مستقر در خانه‌ی خلوت‌تر ارجحیت دارد.

#### اولویت چهارم توپ جنگی:

سرباز مستقر در خانه‌ای که مجموع سلامتی سربازانش کمتر باشد، نسبت به سرباز مستقر در خانه‌ای که سربازان سالم‌تری دارد، ارجحیت دارد.

#### اولویت پنجم توپ جنگی:

سرباز با سلامتی کمتر نسبت به سرباز سالم‌تر اولویت دارد.

در صورتی که شلیک انجام شود، سلامتی تمام سربازانی که مورد هدف قرار گرفته‌اند به اندازه‌ی قدرت تخریب برج کاهش می‌یابد و اگر میزان سلامتی هر یک صفر یا منفی شد، آن سرباز از نقشه‌ی بازی حذف می‌شود و جایزه‌ی کشتن او به بازیکن مدافع اهدا می‌شود. همچنین اگر اولویت چند سرباز یکسان بود، انتخاب به شکل دلخواه و غیرقابل پیش‌بینی انجام می‌شود.

### تصمیم‌گیری بازیکنان:

هر یک از نقش‌های دفاع و حمله دو نوع تصمیم‌گیری بر عهده دارد. یک تصمیم‌گیری در هر نوبت انجام می‌شود و کاری انجام می‌دهد که از نظر پردازشی سبک باشد، تصمیم‌گیری دیگر بعد از گذشتن یک دوره انجام می‌شود و امکان انجام محاسبات سنگین در آن وجود دارد. تصمیم‌هایی که هر نقش می‌تواند بگیرد در ادامه آمده است:

#### فعالیت‌های تیم مهاجم:

- راهی کردن یک سرباز عادی در یک مسیر انتخاب شده
- راهی کردن یک سرباز زره‌پوش در یک مسیر انتخاب شده
- آتش زدن یکی از برج‌ها یا خانه‌های نقشه

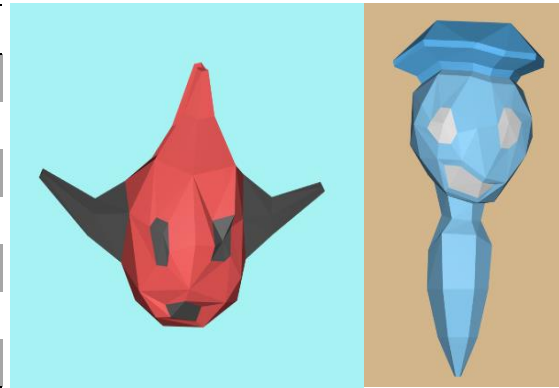




نتیجه‌ی فعالیت‌های راهی کردن قرار گرفتن یک سرباز از نوع عادی یا زره‌پوش در خانه‌ی ابتدایی مسیر انتخاب شده است. ویژگی‌های سرباز قرار گرفته طبق شرایط زیر مشخص می‌شود:

#### ویژگی‌های سرباز عادی تراز ۱:

نام ویژگی	مقدار ویژگی
هزینه ساخت	۴۰
افزایش درآمد	۳
سلامتی اولیه	۳۲
دوره‌ی حرکت	۲
جایزه‌ی کشتن	۸
پتانسیل آسیب	۱
محدوده دید	۲



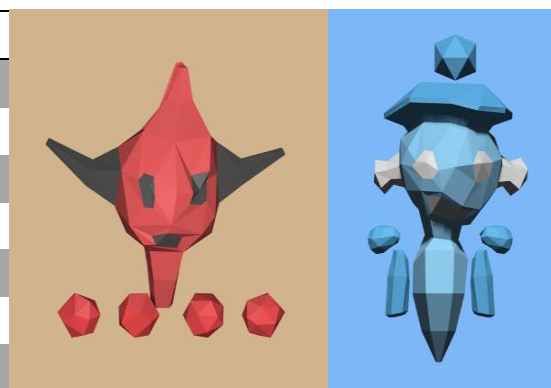
بعد از هر ۶۰ واحد سرباز عادی که ساخته می‌شود، مقدار تراز یک واحد افزایش یافته و ویژگی‌های سرباز جدید بدین شکل تغییر می‌کند:

#### تغییرات ویژگی‌های سرباز عادی به ازای افزایش تراز:

نام ویژگی	تغییر ویژگی
هزینه ساخت	+۱۶
سلامتی اولیه	$\times 1.3$
جایزه‌ی کشتن	+۲

#### ویژگی‌های سرباز زره‌پوش تراز ۱:

نام ویژگی	مقدار ویژگی
هزینه ساخت	۱۸۰
افزایش درآمد	۱۰
سلامتی اولیه	۲۴۰
دوره‌ی حرکت	۴
جایزه‌ی کشتن	۲۵
پتانسیل آسیب	۵
محدوده دید	۲



بعد از هر ۱۵ واحد سرباز زره‌پوشی که ساخته می‌شود مقدار تراز یک واحد افزایش یافته و ویژگی‌های سرباز جدید بدین شکل تغییر می‌کند:







## تغییرات ویژگی‌های سرباز عادی به ازای افزایش تراز:

نام ویژگی	تغییر ویژگی
هزینه ساخت	+۶۰
سلامتی اولیه	$\times 1,3$
جایزه‌ی کشتن	+۸

زمانی که یک بازیکن خانه‌ای از نقشه را آتش می‌زند، خانه تبدیل به خانه مسدود، و اگر برجی از حریف در آن مکان وجود داشت، تخریب می‌شود. **تعداد آتش زنه ها بسته به نقشه متفاوت است.**

## هزینه‌ها و درآمدهای تیم مهاجم:

ساخت سرباز عادی تراز یک به ۴۰ واحد پول نیاز دارد و با هر افزایش تراز این مقدار ۱۶ واحد افزایش پیدا می‌کند. همچنین ساخت سرباز زره‌پوش تراز یک به ۱۸۰ واحد پول نیاز دارد و با هر افزایش تراز این مقدار ۶۰ واحد افزایش پیدا می‌کند.

تیم مهاجم در ابتدای هر نوبت تصمیم‌گیری بزرگ، مقداری پول به عنوان درآمد دریافت می‌کند. این مقدار در ابتدای بازی ۲۰ واحد است و با ساختن هر سرباز عادی به اندازه‌ی ۳ و با ساختن هر سرباز زره‌پوش به اندازه‌ی ۱۰ واحد افزایش پیدا می‌کند.

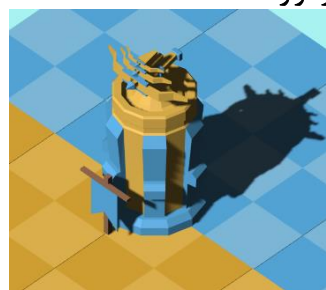
## فعالیت‌های تیم مدافع:

- ساختن یک توپ جنگی در یک خانه‌ی هموار انتخاب شده
- ساختن یک کمان‌دار در یک خانه‌ی هموار انتخاب شده
- ارتقای تراز یک توپ جنگی از قبل موجود
- ارتقای تراز یک کمان‌دار از قبل موجود
- استفاده از رعد و برق

نتیجه‌ی ساختن یک برج، قرارگیری برج با تراز دلخواه در مکان مشخص شده است و نتیجه‌ی ارتقای تراز جایگزینی برج تراز پایین با برج تراز بالاتر می‌باشد. ویژگی‌های برجی که ساخته می‌شود طبق شرایط زیر مشخص می‌شود:

## ویژگی‌های کمان‌دار تراز ۱:

نام ویژگی	مقدار ویژگی
هزینه ساخت	$(5 * \text{تعداد فعلی}) + 125$
هزینه ارتقا	۵۰
قدرت تخریب	۶۰
دوره‌ی شلیک	۴
تیررس	۲





با افزایش یک واحدی مقدار تراز کمان دار ویژگی های آن بدین شکل تغییر می کند:

تغییرات ویژگی های برج کمان دار به ازای افزایش تراز:

نام ویژگی	تغییر ویژگی
هزینه ارتقا	$\times 1.5$
قدرت تخریب	$\times 1.4$

ویژگی های توپ جنگی تراز ۱:

نام ویژگی	مقدار ویژگی
هزینه ساخت	$(5 * \text{تعداد فعلی}) + 175$
هزینه ارتقا	۷۰
قدرت تخریب	۱۰
دوره ی شلیک	۴
تیررس	۲



با افزایش یک واحدی مقدار تراز توپ جنگی ویژگی های آن بدین شکل تغییر می کند:

تغییرات ویژگی های برج توپ جنگی به ازای افزایش تراز:

نام ویژگی	تغییر ویژگی
هزینه ارتقا	$\times 1.5$
قدرت تخریب	$\times 1.4$

در صورت استفاده از رد و برق در یک نقطه از نقشه، تمام سربازهای موجود که با نقطه ی هدف حداکثر ۲ واحد فاصله دارند از بین می روند. معیار فاصله مشابه قبل فاصله ی منهتنی می باشد. **تعداد رد و برق ها بسته به نقشه متفاوت است.** مشابه زمانی که برج ها سربازان را هدف قرار می دهند، کشته شدن سربازان باعث می شود مبلغی به عنوان غنیمت به تیم مدافع داده شود.

هزینه ها و درآمدهای تیم مدافع:

ساخت برج کمان دار **در ابتدا به ۱۲۵ واحد پول و ساخت برج توپ جنگی در ابتدا به ۲۰۰ واحد پول** نیاز دارد. همچنین اولین ارتقای یک کمان دار به ۵۰ واحد پول نیاز دارد و هر ارتقا به میزان ۱،۵ برابر ارتقای قبلی هزینه دارد. اولین ارتقای یک توپ جنگی نیز به ۷۰ واحد پول نیاز دارد و هر ارتقا به میزان ۱،۵ برابر ارتقای قبلی هزینه دارد. همچنین همانطور که ذکر شد کشتن هر سرباز میزان مشخصی سود به تیم مدافع می رساند.



## ترتیب زمانی حوادث:

برای رفع ابهام بار دیگر ترتیب زمانی فعالیت‌هایی که توصیف شد را مرور می‌کنیم.

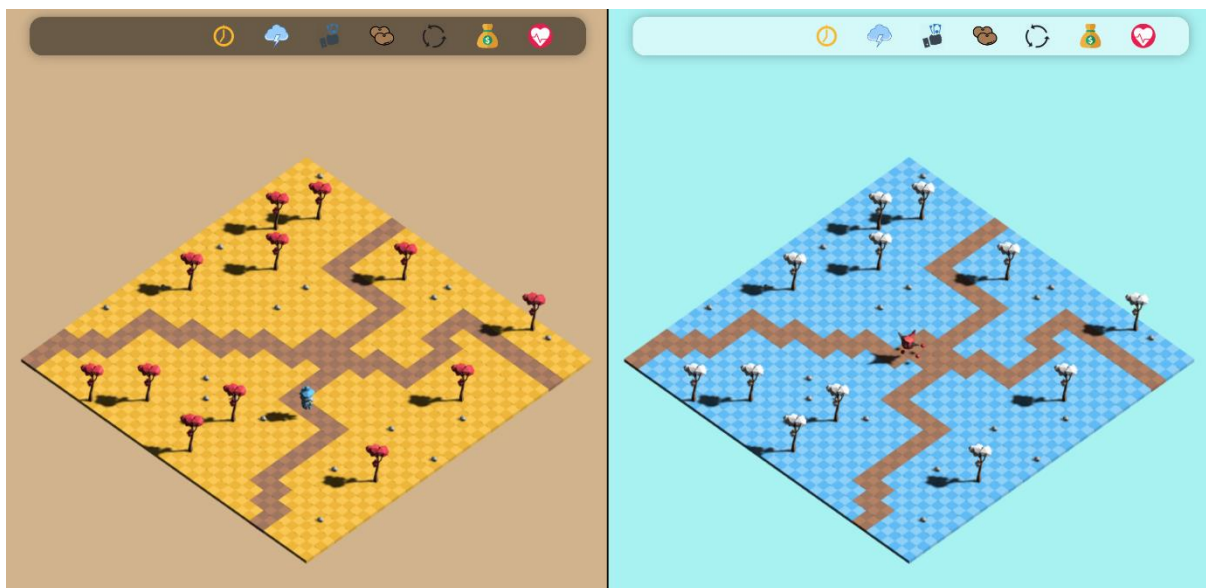
به شکل دوره‌ای از ابتدای بازی قدم تصمیم‌گیری و قدم فعالیت داریم. در قدم تصمیم‌گیری، ابتدا تصمیم‌گیری مربوط به تیم مهاجم و سپس تصمیم‌گیری مربوط به تیم مدافع انجام می‌شود. در قدم فعالیت نیز ابتدا رعد و برق‌ها و آتش زدن برج‌ها و خانه‌ها انجام می‌شود. سپس ابتدا برج‌های مدافع فعالیت کرده و بعد از آن سربازان مهاجم ساخته شده یا حرکت می‌کنند. دوره‌ی تصمیم‌گیری برابر با ۱۰ است، به این معنی که هر بازیکن به ازای هر ۹ قدم تصمیم‌گیری سبک (با محدودیت ۰,۲ ثانیه)، ۱ قدم تصمیم‌گیری با امکان انجام محاسبات پیچیده (با محدودیت ۱ ثانیه) دارد. همچنین بازی زمانی که ۱۰۰۰ قدم فعالیت سپری شده است، به اتمام می‌رسد.

## شروع و پایان بازی:

در ابتدای بازی هر بازیکن ۲۰۰۰ واحد پول دارد و حالت اولیه‌ی نقشه نیز خالی از سربازان است.

**استحکام مقر هر بازیکن بسته به نقشه متفاوت است** و همانطور که قبل‌تر اشاره شد، هر وقت که سربازی به انتهای مسیر می‌رسد مقداری از این استحکام کاهش می‌یابد. هر زمان که استحکام مقر یک بازیکن به ۰ برسد، او بازی را واگذار کرده است. اگر مدت زمان بازی به اتمام برسد و این اتفاق برای هیچ یک از بازیکنان نیفتد، برنده بازیکنی است که استحکام مقر بیشتری داشته باشد. در صورت مساوی بودن استحکام‌ها، بازیکنی برنده است که گردش مالی بیشتری داشته باشد. گردش مالی برابر با سرمایه‌ی فعلی به علاوه‌ی تمام مخارج انجام شده است. در صورت رخداد تساوی در مراحل انتخابی و حضوری، تیمی که کد خود را زودتر ارسال کرده، برنده اعلام می‌شود.

دقت داشته باشید که احتمال دارد قوانین بازی برای مرحله‌ی حضوری تغییرات جزئی داشته باشند.





## شروع کد نویسی (بایدها و نبایدها)

۱. شما باید کد هوش مصنوعی خود را در توابع simpleTurn و complexTurn در فایل AI.java (برای زبان‌های دیگر فایلی به همین نام) قرار دهید.

۲. شما می‌توانید کد کلاینت داده شده را تغییر دهید، به آن فایل اضافه کنید یا از آن فایل حذف کنید، به شرط آن که تغییرات داده شده در کامپایل و اجرای کلاینت و ارتباط آن با سرور اختلالی ایجاد نکند. در مورد هر کلاینت نکاتی ذکر شده که به آن‌ها نیز باید توجه شود. هم‌چنین باید تغییرات احتمالی فایل‌های دیگر کلاینت (فایل‌های غیر از فایلی که در آن کد می‌زنید) را در نظر بگیرید.

۳. شما می‌توانید برای آپدیت بودن کلاینت‌ها یا سرور خود به آخرین ورژن منتشر شده در repository مسابقات مراجعه کنید. این کار کاملاً اختیاری است و اگر به کار با git آشنایی ندارید توصیه نمی‌شود.

<https://github.com/SharifAIChallenge/>

(repo کلاینت‌ها با پسوند AIC18-Client آغاز می‌شود)

## نحوه اجرا و توسعه (کلاینت جاوا)

پیشنهاد ما برای توسعه‌ی کلاینت جاوا استفاده از IntelliJ است. پس از باز کردن IntelliJ گزینه‌ی Import Project را انتخاب کنید و فولدری که کلاینت در آن قرار دارد را انتخاب کنید. در صفحه‌ی Libraries مطمئن شوید gson-2-3-1.jar انتخاب شده است.

## نحوه اجرا و توسعه (کلاینت سی‌پلاس‌پلاس)

برای توسعه‌ی کلاینت سی‌پلاس‌پلاس پیشنهاد ما استفاده از Clion است. پس از نصب Clion در صورتی که از ویندوز استفاده می‌کنید از توضیحات لینک زیر برای تنظیم Clion استفاده کنید:

<https://www.jetbrains.com/help/clion/quick-tutorial-on-configuring-clion-on-windows.html>

پس از نصب و راه‌اندازی Clion کفایت فولدری که پروژه در آن قرار دارد را در Clion باز کنید.

در صورتی که نمی‌خواهید از Clion استفاده کنید نیاز به نصب Cmake دارید. پس از نصب Cmake یک ترمینال باز کنید و به فولدری که پروژه در آن قرار دارد بروید و دستور cmake . را اجرا کنید. پس از این می‌توانید با استفاده از دستور make پروژه را build کنید. بعد از اجرای دستور make فایل باینری client در فولدر جاری ساخته خواهد شد.





## نحوه اجرا و توسعه (کلاينت پايتون)

برای اجرای کلاينت پايتون نیاز به پايتون ۳ دارید. پس از نصب پايتون ۳ کافيسٲ فايل `Controller.py` را اجرا نماييد. شما می‌توانيد به اين کلاينت فايل‌های مورد نظرتان را اضافه کنيد يا فايل‌های آن را تغيير دهيد به شرط آن که کد شما با سرور سازگار باشد و به سادگی بتوانيد ورژن‌های احتمالی بعدی کلاينت را جايگزين نماييد. برای توسعه و کدنویسی به اين زبان، PyCharm را توصیه می‌کنيم.

## ارسال فايل‌ها

برای ارسال فايل‌های خود، فولدر کلاينت را فشرده کنيد و فايل فشرده را ارسال نماييد.





## رابط برنامه نویسی:

نکته مهم: رفرنس‌های موجود در نقشه، در هر نوبت از ابتدا ساخته می‌شوند و رفرنس‌های نوبت قبل معتبر نیستند. (برای

حفظ دسترسی به آپجکت‌ها، توصیه می‌شود از id آن‌ها، که همواره ثابت است، استفاده کنید.)

در ادامه، اجزایی که برای برنامه‌نویسی به آنان نیاز دارید به همراه توضیحی مختصر آمده‌اند.

## Class (or interface) Point

int getX();	مقدار عرض نقطه
int getY();	مقدار ارتفاع نقطه

## Class (or interface) Player

int getMoney();	مقدار پول فعلی بازیکن. پول دشمن را صفر برمی‌گرداند
int getIncome();	مقدار درآمد فعلی بازیکن. درآمد دشمن را صفر برمی‌گرداند
int getStrength();	میزان استحکام فعلی مقرر بازیکن.
int getBeansLeft();	تعداد آتش زنه‌های باقی‌مانده بازیکن
int getStormsLeft();	تعداد رعد و برق‌های باقی‌مانده بازیکن

## enum Owner

ME	خودی
ENEMY	دشمن

## Class (or interface) Entity

int getId();	عدد شناسه موجودیت
Point getLocation();	نقطه قرارگیری موجودیت
Owner getOwner();	مالک موجودیت

## Class (or interface) Unit extends Entity

Path getPath();	مسیر سرباز مورد نظر
int getHealth();	میزان سلامتی. سلامتی سرباز دشمن را صفر برمی‌گرداند
int getLevel();	تراز سرباز مورد نظر
int getPrice();	قیمت خرید سرباز مورد نظر
int getAddedIncome();	میزان درآمد اضافه شده توسط سرباز مورد نظر
int getBounty();	میزان جایزه کشتن سرباز مورد نظر
int getDamage();	میزان پتانسیل آسیب سرباز مورد نظر
int getMoveSpeed();	طول دوره حرکت سرباز مورد نظر
int getVisionRange();	محدوده دید سرباز مورد نظر







#### Class (or interface) HeavyUnit extends Unit

static int INITIAL_PRICE;	قیمت اولیه سرباز
static int PRICE_INCREASE;	میزان افزایش قیمت سرباز به ازای هر تراز
static int INITIAL_HEALTH;	سلامتی اولیه سرباز
static double HEALTH_COEFF;	ضریب افزایش سلامتی سرباز به ازای هر تراز
static int INITIAL_BOUNTY;	جایزه اولیه کشتن سرباز
static int BOUNTY_INCREASE;	میزان افزایش جایزه کشتن سرباز به ازای هر تراز
static int ADDED_INCOME;	میزان افزایش درآمد به ازای ساخت سرباز
static int MOVE_SPEED;	طول دوره حرکت سرباز
static int DAMAGE;	میزان پتانسیل آسیب سرباز
static int VISION_RANGE;	محدوده دید سرباز
static int LEVEL_UP_THRESHOLD;	تعداد سربازهای زره پوش لازم برای ارتقا به تراز بالاتر

#### Class (or interface) LightUnit extends Unit

static int INITIAL_PRICE;	قیمت اولیه سرباز
static int PRICE_INCREASE;	میزان افزایش قیمت سرباز به ازای هر تراز
static int INITIAL_HEALTH;	سلامتی اولیه سرباز
static double HEALTH_COEFF;	ضریب افزایش سلامتی سرباز به ازای هر تراز
static int INITIAL_BOUNTY;	جایزه اولیه کشتن سرباز
static int BOUNTY_INCREASE;	میزان افزایش جایزه کشتن سرباز به ازای هر تراز
static int ADDED_INCOME;	میزان افزایش درآمد به ازای ساخت سرباز
static int MOVE_SPEED;	طول دوره حرکت سرباز
static int DAMAGE;	میزان پتانسیل آسیب سرباز
static int VISION_RANGE;	محدوده دید سرباز
static int LEVEL_UP_THRESHOLD;	تعداد سربازهای عادی لازم برای ارتقا به تراز بالاتر

#### Class (or interface) Tower extends Entity

int getLevel();	تراز فعلی برج
int getPrice();	مجموع هزینه های فعلی برج. برای برج های حریف صفر برمی گرداند
int getDamage();	قدرت تخریب فعلی برج
int getAttackSpeed();	دوره شلیک برج
int getAttackRange();	تیررس برج





## Class (or interface) CannonTower extends Tower

static int INITIAL_PRICE;	قیمت اولیه برج
static int INITIAL_PRICE_INCREASE;	افزایش قیمت اولیه به ازای هر برج جدید از این نوع
static int INITIAL_LEVEL_UP_PRICE;	قیمت اولیه ارتقای برج
static double PRICE_COEFF;	ضریب قیمت ارتقای برج به ازای هر تراز
static int INITIAL_DAMAGE;	قدرت تخریب اولیه برج
static double DAMAGE_COEFF;	ضریب افزایش قدرت تخریب برج به ازای هر تراز
static int ATTACK_RANGE;	تیررس برج
static int ATTACK_SPEED;	طول دوره شلیک برج

## Class (or interface) ArcherTower extends Tower

static int INITIAL_PRICE;	قیمت اولیه برج
static int INITIAL_PRICE_INCREASE;	افزایش قیمت اولیه به ازای هر برج جدید از این نوع
static int INITIAL_LEVEL_UP_PRICE;	قیمت اولیه ارتقای برج
static double PRICE_COEFF;	ضریب قیمت ارتقای برج به ازای هر تراز
static int INITIAL_DAMAGE;	قدرت تخریب اولیه برج
static double DAMAGE_COEFF;	ضریب افزایش قدرت تخریب برج به ازای هر تراز
static int ATTACK_RANGE;	تیررس برج
static int ATTACK_SPEED;	طول دوره شلیک برج

## Class (or interface) Cell

Point getLocation();	نقطه مختصاتی خانه مورد نظر
----------------------	----------------------------

## Class (or interface) RoadCell extends Cell

ArrayList<Unit> getUnits();	لیست سربازهای مستقر در خانه مورد نظر، می تواند تهی باشد
-----------------------------	---

## Class (or interface) GrassCell extends Cell

Tower getTower();	برج قرار گرفته در خانه مورد نظر، می تواند هیچ مقدار باشد
-------------------	--

## Class (or interface) BlockCell extends Cell





## Class (or interface) Map

<code>int getWidth();</code>	عرض نقشه
<code>int getHeight();</code>	ارتفاع نقشه
<code>Cell[][] getCellsGrid();</code>	آرایه دوبعدی از خانه های نقشه
<code>Cell getCell(int x, int y);</code>	خانه موجود در مختصات مورد نظر
<code>ArrayList&lt;Cell&gt; getCellsList();</code>	لیست خانه های نقشه

## Class (or interface) Path

<code>ArrayList&lt;RoadCell&gt; getRoad();</code>	لیست خانه های مسیر
---	--------------------

## Class (or interface) GameEvent

<code>Owner getOwner();</code>	مالک رویداد رخ داده شده (رعد و برق/ آتش زدن)
<code>Point getLocation();</code>	نقطه مکان رخداد رویداد

## Class (or interface) StormEvent extends GameEvent

## Class (or interface) BeanEvent extends GameEvent

## Class (or interface) Game implements World

<code>static int INITIAL_STRENGTH;</code>	میزان استحکام اولیه مقر
<code>static int MAX_TURNS_IN_GAME;</code>	حداکثر نوبت های بازی
<code>static int INITIAL_MONEY;</code>	مقدار پول اولیه
<code>static int INITIAL_BEANS_COUNT;</code>	تعداد آتش زنه های اولیه
<code>static int INITIAL_STORMS_COUNT;</code>	تعداد رعد و برق های اولیه
<code>static int STORM_RANGE;</code>	محدوده برخورد رعد و برق

## Class (or interface) AI

<code>void simpleTurn(World world);</code>	تابع محاسبات سبک که در همه نوبت ها به جز مضارب ۱۰ اجرا می شود. محدودیت زمانی ۰,۲ ثانیه
<code>void complexTurn(World world);</code>	تابع محاسبات سنگین که فقط در نوبت های مضارب ۱۰ اجرا می شود. محدودیت زمانی ۱ ثانیه





Class (or interface) World

<code>int getCurrentTurn();</code>	نوبت فعلی بازی
<code>ArrayList&lt;Unit&gt; getMyUnits();</code>	لیست سربازهای زنده فعلی خودی
<code>ArrayList&lt;Unit&gt; getEnemyUnits();</code>	لیست سربازهای زنده فعلی دشمن. سربازهای دشمن همگی قابل مشاهده هستند.
<code>ArrayList&lt;Tower&gt; getMyTowers();</code>	لیست برج های فعلی خودی
<code>ArrayList&lt;Tower&gt; getVisibleEnemyTowers();</code>	لیست برج های تاکنون دیده شده دشمن
<code>Map getAttackMap();</code>	نقشه ای که بازیکن در آن، مهاجم است
<code>Map getDefenceMap();</code>	نقشه ای که در آن بازیکن در آن، مدافع است
<code>ArrayList&lt;Path&gt; getAttackMapPaths();</code>	لیست مسیرهای نقشه ای که بازیکن در آن، مهاجم است
<code>ArrayList&lt;Path&gt; getDefenceMapPaths();</code>	لیست مسیرهای نقشه ای که بازیکن در آن، مدافع است
<code>Player getEnemyInformation();</code>	اطلاعات بازیکن دشمن
<code>Player getMyInformation();</code>	اطلاعات بازیکن خودی
<code>ArrayList&lt;Unit&gt; getDeadUnitsInThisTurn();</code>	لیست سربازهای که در این نوبت کشته شده اند
<code>ArrayList&lt;Unit&gt; getPassedUnitsInThisTurn();</code>	لیست سربازهایی که در این نوبت مسیر خود را به اتمام رساندند
<code>ArrayList&lt;Tower&gt; getDestroyedTowersInThisTurn();</code>	لیست برج هایی که در این نوبت نابود شده اند
<code>ArrayList&lt;BeanEvent&gt; getBeansInThisTurn();</code>	لیست آتش زنه هایی که در این نوبت اتفاق افتاده اند
<code>ArrayList&lt;StormEvent&gt; getStormsInThisTurn();</code>	لیست رعد و برق هایی که در این نوبت اتفاق افتاده اند
<code>void createLightUnit(int pathIndex);</code>	ساخت سرباز عادی در مسیر با اندیس مورد نظر
<code>void createHeavyUnit(int pathIndex);</code>	ساخت سرباز زره پوش در مسیر با اندیس مورد نظر
<code>bool isTowerConstructable(int x, int y);</code>	اگر در خانه مختصات مورد نظر بتوان برج ساخت، درست برمی گرداند.
<code>void createCannonTower(int level, int x, int y);</code>	ساخت برج توپ جنگی. در صورت کمبود بودجه مورد نیاز، حداکثر تراز ممکن در نظر گرفته می شود
<code>void createArcherTower(int level, int x, int y);</code>	ساخت برج کماندار. در صورت کمبود بودجه مورد نیاز، حداکثر تراز ممکن در نظر گرفته می شود
<code>void upgradeTower(Tower tower);</code>	ارتقای تراز برج مورد نظر
<code>void plantBean(int x, int y);</code>	آتش زدن خانه مورد نظر
<code>void createStorm(int x, int y);</code>	ایجاد رعد و برق حول خانه مورد نظر

