



SHARIF AI CHALLENGE

2020



داک بازی AI20

داستان بازی

در دوران دوم زندگانی مردم در زمین، پادشاهان و سلاطین مختلفی در ایالات مختلفی زندگی کرده و مردم هم در آسایش بودند؛ همکاری بین پادشاهان به جایی رسید که برای تجارت بهتر و گسترش همکاری‌ها راه‌ها و جاده‌های مختلفی را بین ایالات خود کشیدند و باعث پیشرفت فناوری در کشور های هم شدند. اما در دوران سوم در میان همه‌ی پادشاهان و سلاطین حرص ، نفس و جاه‌طلبی بزرگترین پادشاهان بودند به صورتی که تمامی سلاطین را برای گرفتن بقیه‌ی ایالت‌ها به جان هم انداخت. سلاطین برای بدست آوردن ایالات دیگر ناگزیر از همکاری با بقیه‌ی کشور ها بودند و برای اینکه بهترین همکار را بدست آورند چندین یار را امتحان می‌کنند تا با یاران قوی‌تر ایالات بیشتری کسب کنند. بنابراین همه‌ی پادشاهان با یارانشان در جنگ حاضر می‌شدند و از راه هایی بین ایالات مختلف بود استفاده می‌کردند تا یا به یکدیگر کمک کنند یا با کمک هم دشمنان را نابود کنند. حال شما سپه‌سالاران این پادشاهان هستید و وظیفه‌ی راهی کردن یگان ها و سربازان مختلف را برعهده دارید. به امید پیروزی ...





AI CHALLENGE

2020

عناصر بازی

تیم‌ها: هر بازی شامل ۴ تیم می شود که به صورت ۲ به ۲ در مقابل هم قرار گرفته و بازی می کنند. در واقع در هر بازی کد شما یاری از جنس کد و هوش مصنوعی دارد که باید برای رسیدن به پیروزی با هم همکاری کنید.

نقشه: هر بازی در یک نقشه که شامل عناصر زیر است انجام می شود :

- **تایل:** به هر خانه از نقشه یک تایل گفته می شود.
- **شاه (King) :** خانه‌ی شاه خانه‌ای 3×3 در نقشه می باشد که قابلیت حمله به حریفان را دارد. هر شاه بردی دارد که از تایل وسط آن در نظر گرفته می شود؛
- **مسیر :** هر مسیر دنباله‌ای با عرض یک از تایل‌ها می باشد. هر مسیر از یک خانه‌ی شاه شروع و به خانه‌ی شاه دیگری ختم می شود. (بین دو شاه دوست تنها یک مسیر یکتا وجود خواهد داشت) تمامی راه‌ها از وسط این شاه‌ها آغاز شده و تا تایل وسط از شاه‌های دیگر ادامه دارد. این راه‌ها می توانند overlap داشته باشند.
- **فاصله:** فاصله‌ی یک تایل با تایل دیگر مجموع قدر مطلق اختلاف ستون‌های دو تایل و قدر مطلق اختلاف سطرهای دو تایل می باشد. (فاصله‌ی منتهی)

یونیت‌ها: یونیت‌ها سربازان شما هستند که با سرعت هایی یکسان در بازی حرکت می کنند. سربازانی که ویژگی‌های زیر را دارا می باشند :

- **مکان (Location):** هر سرباز در یک تایل جای می گیرد؛ حال میتواند در این تایل در هوا باشد یا روی زمین.



AI CHALLENGE

2020

- سلامتی (HP): میزان سلامتی معیار سنجیده شدن جان و سلامتی یک سرباز است که اگر کوچکتر مساوی ۱ باشد آن سرباز مرده محسوب می شود.
 - ضربه (Damage): میزان ضربه‌ی که یک سرباز به سرباز حریف ، میزانی است که از سلامتی سرباز حریف می‌کاهد.
 - هدف (Target): هدف هر سرباز، در واقع مکان سربازانی است که می‌تواند به آنها را ضربه بزند: به عنوان مثال اگر هدف یک سرباز هوا باشد تنها می‌تواند سربازان حریف را که در هوا هستند مورد ضربه قرار دهد یا اگر هدف هم زمین و هم هوا باشد ، می‌تواند تمامی سربازان حریف چه در هوا باشند چه در زمین را بزند. از طرفی هر یونیت می‌تواند یک یونیت در یک تایل را بزند (Single) یا کل یونیت‌های داخل یک تایل را مورد حمله قرار دهد (Multiple)
 - امتیاز عمل (AP): شما یک میزان امتیاز عمل کلی دارید که در هر نوبت میزانی به آن افزوده می شود. حال برای استفاده از هر سرباز باید معادل امتیاز عمل مورد نیاز آن سرباز خرج کنید.
 - برد (Range) : هر سرباز بردی برای ضربه زدن به حریف دارد که از جنس فاصله از سرباز یا شاه حریف می‌باشد.
 - اولویت ضربه: یونیت ها در اختیار بازیکن ها نیستند؛ اولویت هدف ضربه زدن هر یونیت با یونیتی از حریف است که به ترتیب به آن نزدیک تر و یا جانش کمتر است یا میزان دمج آن بیشتر است؛ در نهایت اگر همه این ملاک ها در دو یونیت برابر باشد، یونیت با آیدی کمتر به عنوان هدف انتخاب می شود. این هدف تا زمانی که زنده است و در دامنه ضربه زدن قرار دارد مورد حمله‌ی یونیت قرار می‌گیرد. همین رویکرد در مورد شاه نیز به همین شکل است.
- اسپل (Spell): اسپل‌ها، عناصری هستند که با استفاده‌ی آنها روی تایل‌هایی که مدنظر دارید می‌توانید اثراتی روی یونیت‌های خودی (هم یونیت های خودتان ، هم یارتان) و یونیت های دشمن بگذارید.
- دک (Deck): مجموعه سربازانی که در طول بازی می‌توانید استفاده کنید را گویند.



AI CHALLENGE

2020

هند (Hand): سربازانی که در نوبت فعلی می‌توانید از آنها استفاده کنید مجموعه‌ای را تشکیل می‌دهند به نام هند که با استفاده از هر سرباز آن سرباز از هند خارج می‌شود و سربازی دیگر که در هند نیست و در دک هست به هند اضافه می‌شود.

امتیاز عمل هر بازیکن (MaxAP): هر بازیکن در بازی از یک میزان امتیاز عمل برخوردار است که آنها را برای استفاده از یونیت‌های مختلف خرج می‌کند

روند بازی:

شروع بازی :

در این بازی ۴ بازیکن وجود دارد که به صورت ۲ به ۲ بازی می‌کنند؛ هر بازیکن تنها بر روی شاه خود تسلط دارد؛ تعداد این راه‌ها برای هر زوج شاه یکسان خواهد بود. بازیکن‌ها با قرار دادن یونیت‌های خود در این راه‌ها از طریق شاه خود و راهی کردن آن‌ها به سمت شاه‌های حریف ، آن‌ها را تهدید می‌کنند. هنگامی که یک شاه نابود شد راه‌های منتهی به آن از طریق راه متصل به شاه یار ، به شاه یار متصل می‌شوند.

"هند" اینگونه تعیین می‌شود که بازیکن ابتدا ۵ یونیت را انتخاب می‌کند که در دست اولیه باشند؛ با بازی کردن با هر یونیت ، آن یونیت از دست خارج شده و یکی از یونیت‌های دیگر که داخل دست نیستند به صورت تصادفی وارد دست می‌شود. وارد شدن به صورت تصادفی نیز به شکل وزن‌دار خواهد بود؛ یعنی از هر یونیت که کمتر استفاده شده باشد ، احتمال اینکه وارد دست شود بیشتر می‌شود. (وزن هر یونیت = معکوس تعداد باری که از یونیت استفاده شده است به علاوه ۱)

اطلاعات بازیکن از بازی :

هر بازیکن می‌داند که یونیت‌هایی که استفاده کرده است در کجای مپ و روی کدام راه هستند. هر یونیت یا حرکت می‌کند یا اگر توانایی ضربه زدن به شاه با سرباز حریف (بر حسب برد و هدف سرباز) را داشت، ضربه می‌زند. به عبارت دیگر یونیت‌هایی که در راه‌ها راهی شدند اگر به یونیت دشمن برخورد کردند بر حسب اینکه در هوامی باشد یا در زمین در صورت توانایی با آنها مبارزه می‌کنند یا از آنها عبور می‌کنند و در انتها ی راه به شاه حریف می‌رسند و به شاه صدمه وارد می‌کنند تا از میزان سلامتی آن کاسته شود.

هر شاه برای دفاع از خود تا برد معینی را می‌تواند اтак بدهد. این برد برابر ۵ می‌باشد که از وسط آن و به صورت منتهی حساب می‌شود.



AI CHALLENGE

2020

هر Agent می‌تواند این اطلاعات را از یار خود ببیند که در این ترن چه یونیتی را بر روی چه راهی قرار داده است و الان چه اسپلی (اگر دارد) در اختیار دارد. هر Agent از حریفانش این را می‌داند که چه یونیتی را روی زمین گذاشته است ولی نمی‌داند در چه راهی گذاشته است.

تعامل با اسپل‌های بازی:

هر ۱۰ ترن ۲ اسپل متفاوت به صورت رندوم از مجموعه‌ی اسپل‌ها انتخاب شده و به صورت یکسان به تیم‌ها داده می‌شود؛ به عنوان مثال اسپل‌های Heal و Haste انتخاب شده و به هر دو تیم داده می‌شود. سپس به صورت رندوم در اختیار هر کدام از Agent‌های هر تیم یکی از آن اسپل‌ها قرار داده می‌شود و می‌توانند بعد از آن هر موقع که دلخواهشان بود استفاده کنند.

نحوه ی گذشتن یک نوبت :

یک نوبت به این ترتیب انجام می‌شود که بازیکن اسپلی را (اگر بخواهد) استفاده می‌کند و یونیتی را (اگر بتواند و بخواهد) روی راهی مشخص می‌گذارد. سپس یونیت‌ها کارهایی که می‌توانند را انجام می‌دهند (یعنی اگر هدف قابل ضربه زدنی داشت ، ضربه می‌زند اما اگر نداشت با توجه به سرعتش بر روی راه مشخص خود حرکت می‌کند)

نکات تکمیلی در رابطه با راه‌ها :

شاه‌ها به صورت مربع‌های ۳×۳ هستند که راه‌ها از تایل وسطی آنها آغاز و به تایل وسط مربع شاه دیگر ختم می‌شود.

راه‌ها می‌توانند با هم هم‌پوشانی داشته باشند و یک راه از قسمتی از راه دیگر رد شود. توجه داشته باشید که برد یونیت‌ها ورای راه‌ها می‌باشد و در صورتی که یونیتی از حریف در راه دیگر و در برد یونیت شما باشد، آن یونیت را به عنوان هدف خواهد شناخت و تا زمانی که یکی از یونیت‌ها از بین بروند یا از برد هم خارج شوند به آن ضربه خواهد زد.

در طول بازی ممکن است که شاه‌ها منهدم شوند در این صورت راه‌های منتهی به آن شاه از طریق راه یکتایی که بین دو شاه یار هست به تایل وسط شاه یار آن متصل می‌شود و در صورت وجود یونیت روی راه‌های منتهی به شاه منهدم شده، آن یونیت‌ها از طریق راه یکتای مذکور به شاه دیگر حریف راه پیدا می‌کنند (توضیحات داده شده روی راه‌ها و یونیت‌ها هستند که توسط بازیکن کنترل نمی‌شوند و این توضیحات جنبه‌ی پیاده‌سازی ندارند بلکه قوانین طبیعی بازی محسوب می‌شوند)

از آنجایی که شاه‌ها از ۹ تایل ساخته می‌شوند، در هر راه در صورتی که هر تایل از شاه‌ها در برد یونیت قرار بگیرد آن شاه مورد ضربه‌ی یونیت قرار خواهد گرفت.

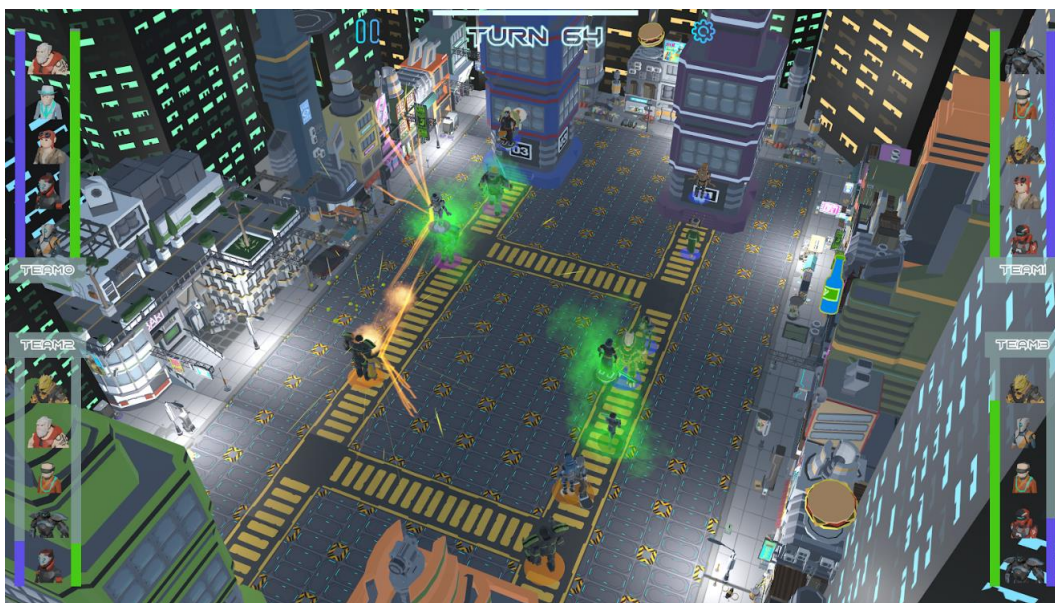


AI CHALLENGE 2020

رتبه‌بندی :

در آخر تیمی برنده می شود که مجموع سلامتی شاه هایش بیشتر باشد. تیم برنده و بازنده در مجموع به ترتیب ۱۴ امتیاز و ۶ امتیاز دریافت می‌کنند؛ در تیم برنده بازیکنی که شاهش از سلامت بیشتری برخوردار باشد ۸ و بازیکن دیگر تیم ۶ و در تیم بازنده بازیکنی که شاهش میزان بیشتری سلامتی داشته باشد ۴ و دیگری ۲ امتیاز دریافت می‌کند. اگر در انتهای بازی در هر تیم سلامتی شاه‌ها یکسان بود امتیاز تیم بین اعضا تقسیم می شود. اگر بازی مساوی شود به هر تیم ۱۰ امتیاز می‌دهیم و به بازیکن با سلامتی بیشتر ۶ امتیاز و به بازیکن با سلامتی کمتر ۴ امتیاز می‌دهیم هر بازیکن در نهایت بر اساس امتیازی که در بازی‌ها گرفته است رتبه‌بندی می شود.

جزئیات اسپل‌های بازی:



این اسپل‌ها می‌توانند یک منطقه‌ی مربعی را مورد تاثیر قرار دهند؛ بازیکنانی‌توانند با نشانه کردن یک tile در جهت ۸ خانه‌ی همسایه اش تاثیر مدنظر را بگذارند. یا می‌تواند روی یک تایل این اسپل را بزند. تمامی اسپل‌ها حداکثر تا ۳ ترن تاثیر می‌گذارند و در هر ترن فقط ۱ اسپل توسط هر بازیکن قابل استفاده می‌باشد. (شایان ذکر است که برای اسپل‌هایی که بیش از ۱ ترن طول می‌کشند تاثیر تنها بر روی یونیت‌هایی است که موقع استفاده از اسپل در محدوده ی آن بوده اند)

ترتیب اثر هر اسپل هم به صورت زیر است:

heal -۱

damage/poison -۲

tele -۳

haste -۴

duplicate -۵



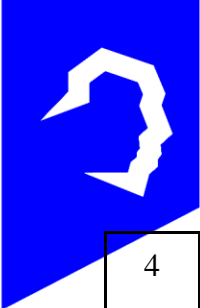
AI CHALLENGE

2020

| Name | Detail | Duration (turn) |
|-----------|--|-------------------|
| Damage | به تمامی یونیت‌های حریف که در آن tile و ۸ همسایه‌اش هست یک میزان خاصی ضربه می‌زند | 1 |
| Heal | تمام HP یونیت‌های خودی در آن tile و ۸ همسایه‌اش ۲ تا افزایش می‌یابد | 1 |
| Tele | از بین نیمه‌ی اول همه راه‌ها می‌توان یونیت‌های خودی یک تایل رو به آنها منتقل کند | 1 |
| Duplicate | از روی یونیت‌های یک تایل دو یونیت کپی می‌سازد که از نظر Damage و HP یک‌چهارم است | 3 |
| Poison | به تمامی یونیت‌های حریف که در آن tile و ۸ همسایه‌اش هست یک میزان خاصی سم می‌زند (که این سم هر ترن میزان خاصی ضربه وارد می‌کند) | 3 |
| Haste | سرعت تمام یونیت‌های خودی در آن tile و ۸ همسایه‌اش حداکثر به ۲ tile بر turn افزایش می‌یابد | 2 |

جزئیات یونیت‌های بازی :

| id | Location | Range | Damage | Target | HP | AP |
|----|----------|-------|--------|-----------------------|----|----|
| 0 | Ground | ۴ | 15 | Single Ground/ Air | 15 | 4 |
| 1 | Ground | ۴ | 5 | Multiple Ground | 30 | 3 |
| 2 | Ground | ۴ | 10 | Multiple Air | 20 | 4 |
| 3 | Ground | ۱ | 12 | Single Ground | 25 | 6 |



AI CHALLENGE

2020

| | | | | | | |
|---|--------|---|----|--------------------|----|----|
| 4 | Ground | ۱ | 10 | Multiple Ground | 50 | 10 |
| 5 | Air | ۱ | 5 | Single Air | 30 | 3 |
| 6 | Air | ۴ | 5 | Single Ground/ Air | 15 | 2 |
| 7 | Air | ۲ | 8 | Multiple Ground | 30 | 5 |
| 8 | Air | ۲ | 15 | Single Ground/ Air | 20 | 6 |

مشخصات کلی بازی :

- امتیاز عمل اولیه : ۱۰
- تعداد ترن بازی : ۱۰۰
- تعداد ترن برای آمدن اسپل‌های بعدی : ۱۰
- سائز هند : ۵
- سائز دک : ۹
- امتیاز عمل اضافه‌شده در هر ترن : ۲
- شاه‌ها :
- سلامتی : ۹۰
- قدرت ضربه : ۱۰
- برد : ۵

دستورات کمکی محیط Command Line:

- با دستور زیر میتوانید از طریق Command Line نقشه مورد نظر خود را به عنوان ورودی پیش از اجرا به سرور بدهید:

```
export AICMap=/path/to/map
```

در ویندوز از طریق environment variable یا با استفاده از دستور set یا setx این کار امکان پذیر است.

```
setx AICMap /path/to/map
```

- با آپشن extra میتوان به محدودیت زمانی های بازی اضافه کرد. برای مثال با دستور زیر ۴۰۰ میلی ثانیه به تایم اوت هر ترن اضافه می شود.

```
java -jar server.jar --extra=400
```



AI CHALLENGE 2020

شروع کدنویسی (بایدها و نبایدها)

۱. شما باید کد هوش مصنوعی خود را در توابع pick و turn و end در فایل Al.java (برای زبان های 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 آنکه 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 فایل های دیگر کلاینت (فایل هایی غیر از فایلی که در آن کد میزنید) را در نظر بگیرید.
 ۳. شما میتوانید برای به روز بودن کلاینت ها یا سرور خود به آخرین نسخه منتشر شده در repository مسابقات مراجعه کنید.
- این کار کاملاً اختیاری است و اگر به کار با git آشنایی ندارید توصیه نمی شود.
- <https://github.com/SharifAIChallenge/>
- (ریپو کلاینت ها با پیشوند AIC20-Client آغاز می شود)

نحوه اجرا بازی به صورت لوکال

۱. ابتدا فایل جر سرور باید ران شود.
۲. سپس یک پنجره باز می شود که فایل مپ باید به آن ورودی داده شود. برای خودکارسازی این امر از وارد کردن مسیر فایل مپ در environment variable که در بالا اشاره شد استفاده کنید.
۳. پس از آن با ران کردن 4 کلاینت به صورت همزمان بازی اجرا می شود. توجه کنید که ترتیب ران کردن کلاینت ها همتمیمی ها را مشخص میکند. کلاینت اول و سوم با هم و کلاینت دوم و چهارم با هم همتمیمی میشوند.
۴. برای توضیحات تکمیلی برای ران کردن کلاینت به فایل readme آنها مراجعه کنید.

نکات فنی

- همه فیلدهای کلاس ها که در زیر آمده اند در کلاینت پایتون به طور مستقیم و در کلاینت های دیگر توسط getter شان قابل دسترسی میباشند.
- در صورت نامعتبر بودن یک دستور، این دستور نادیده گرفته میشود و اولین دستور معتبر بعدی انجام میشود. ترتیب دستورات به همان ترتیبی است که شما در کلاینت ارسال میکنید.
- در صورتی که یک حرکت یا قابلیت به هر دلیل انجام نشود، امتیاز عمل آن از تیم کسر نمیشود.
- رفرنس آجکت world در توابع کلاس AI هر ترن از ابتدا ساخته میشوند و رفرنس های نوبت قبل معتبر نیستند.



AI CHALLENGE

2020

- اکثر توابعی که ورودی‌های spell, unit, player و cell دارند، با ورودی‌های شناسه و row, col هم پیاده‌سازی شده‌اند.
- در مورد بالا در کلاینت پایتون، متدها به صورت دیگری مقداردهی می‌شوند. برای مثال فرض کنید در یکجا Cell و در جای دیگر ورودی‌های row و col داده شوند. نحوه نگارش به صورت زیر خواهد بود:

```
method_name( Cell = ... , ... )  
method_name (row = ... , col = ..., , ... )
```

- توابعی که اطلاعات را از کلاینت به سرور می‌فرستد، روی اطلاعات همان ترن کلاینت تاثیری نمی‌گذارند. مثلاً با صدا زدن تابع putUnit یونیت مورد نظر همان موقع از دست خارج نمی‌شود و میزان ap بازیکن کم نمی‌شود؛ بلکه در ترن بعد این اطلاعات آپدیت خواهد شد. بنابراین کدی مثل
`While(myPlayer.getAp() > 3) {world.putUnit(...);}`

ممکن است هیچ وقت به پایان نرسد و در نتیجه دستورات بعدی آن ترن اجرا نشوند. در صورت تمایل می‌توانید خودتان فیلد ap را برای بازیکن ست کنید (دسترسی وجود دارد) تا در ترن فعلی استفاده کنید اما توجه داشته باشید که ترن بعد رفرنس‌ها عوض میشوند و فیلدها آپدیت خواهند شد.

- کلمات attack و damage از لحاظ مفهومی معادلند. برای مثال آپگرید کردن دمیج یک یونیت، فیلد attack آن را افزایش میدهد.
- یونیت پایه را نمی‌توان به طور صریح روی مسیر بین دو یار قرار داد. برای اینکه یونیت را از یار عبور دهید باید از مسیرهای متصل به یار (به جز مسیر بین دو یار) استفاده کنید؛ در این صورت یونیت قرار داده شده از مسیر بین دو یار عبور می‌کند و پس از رسیدن به شاه یار ادامه مسیرش را از مسیر انتخاب شده می‌رود. همچنین در صورتی که شاهی بمیرد، یونیت‌ها ادامه مسیرشان را از مسیر بین دو یار ادامه می‌دهند تا به شاه دیگر حمله را ادامه دهند. در تمام طول مسیر، مسیر یونیت مسیر انتخاب شده را نشان میدهد.



AI CHALLENGE

2020

AI

```
Void pick(World);
```

این تابع در ابتدای بازی صدا می شود - برای انجام پیش‌پردازش بازی و انتخاب دک

```
Void turn(World);
```

این تابع هر ترن یک بار صدا می شود - برای انجام حرکات مورد نیاز بازی

```
Void end(World, scores);
```

این تابع تنها در انتها پس از پایان بازی صدا زده می شود و اعمال انجام شده در این تابع روی بازی تاثیری نمیگذارد. با استفاده از این تابع به نتیجه بازی دسترسی وجود دارد و از آن میتوان برای آموزش هوش مصنوعی و پردازش‌های پس از بازی استفاده کرد. ورودی scores یک مپ از int به int است که کلید آن آیدی بازیکن و مقدار آن امتیاز بازیکن است.

World

```
Void chooseDeckById(typeIds: List[int])
```

```
Void chooseDeck(baseUnits: List[baseUnit])
```

انتخاب دک در ابتدای بازی، یونیت پایه های اول در لیست به عنوان هند انتخاب می‌شوند.

```
Player getMe()
```

```
Player getFriend()
```

```
Player getFirstEnemy ()
```

```
Player getSecondEnemy()
```

بازیکن های بازی را برمیگرداند

```
Map getMap()
```

مپ بازی را برمیگرداند



AI CHALLENGE

2020

| |
|---------------------------------------|
| List[Path] getPathsCrossingCell(Cell) |
|---------------------------------------|

| |
|---|
| List[Path] getPathsCrossingCell(row, col) |
|---|

| |
|--|
| تمام مسیرهایی که از خانه داده شده رد می‌شوند (شامل آن خانه هستند) را برمیگرداند. |
|--|

| |
|-------------------------------|
| List[Unit] getCellUnits(Cell) |
|-------------------------------|

| |
|-----------------------------------|
| List[Unit] getCellUnits(row, col) |
|-----------------------------------|

| |
|---|
| یونیت های داخل خانه داده شده را برمیگرداند. |
|---|

| |
|--|
| Path getShortestPathToCell(Player, Cell) |
|--|

| |
|--|
| Path getShortestPathToCell(Player, row, col) |
|--|

| |
|--|
| با ورودی گرفتن یک بازیکن و یک خانه کوتاه ترین مسیر از آن بازیکن به آن خانه را برمیگرداند. مسیر میتواند از همتیمی عبور کند. |
|--|

| |
|------------------------------|
| void putUnit(BaseUnit, Path) |
|------------------------------|

| |
|---|
| یونیت داده شده را روی مسیر داده شده راهی میکند. |
|---|

| |
|----------------------|
| int getCurrentTurn() |
|----------------------|

| |
|--------------------|
| ترن فعلی را میدهد. |
|--------------------|

| |
|------------------------|
| int getRemainingTime() |
|------------------------|

| |
|---|
| زمان باقی مانده تا تایم اوت را به میلی ثانیه میدهد. |
|---|

| |
|---|
| Void castUnitSpell(Unit, Path, Cell, Spell) |
|---|

| |
|---|
| Void castUnitSpell(Unit, Path, row, col, Spell) |
|---|

| |
|---|
| برای اسپل تلپورت یونیت داده شده را در زمین به مسیر داده شده و به خانه داده شده میبرد. |
|---|

| |
|---------------------------------|
| Void castAreaSpell(Cell, Spell) |
|---------------------------------|

| |
|-------------------------------------|
| Void castAreaSpell(row, col, Spell) |
|-------------------------------------|

| |
|--|
| روی خانه داده شده اسپل محیطی را کست میکند. |
|--|

| |
|---|
| List[Unit] getAreaSpellTargets(Cell, Spell) |
|---|

| |
|---|
| List[Unit] getAreaSpellTargets(row, col, Spell) |
|---|

| |
|---|
| این تابع میگوید اگر اسپل محیطی روی خانه ای کست شود به چه یونیت هایی میخورد. |
|---|



AI CHALLENGE

2020

```
int getRemainingTurnsToUpgrade()
```

```
int getRemainingTurnsToGetSpell()
```

به ترتیب ترن های باقی مانده تا دریافت توکن آپگرید و اسپل را میدهد. در ترنی که توکن یا اسپل داده می شود مقدار این تابع ماکزیمم خود است؛ به عبارت دیگر خروجی تابع هیچ وقت صفر نیست.

```
int getRangeUpgradeNumber()
```

```
int getDamageUpgradeNumber()
```

به ترتیب تعداد توکن های آپگرید رنج و دمیج قابل استفاده را میدهند.

```
Spell getReceivedSpell()
```

اسپلی که در این ترن به دستمان رسیده را میدهد. اگر اسپل این ترن داده نشده باشد، خروجی null است.

```
Spell getFriendReceivedSpell()
```

اسپلی که در این ترن به دست هم تیمی رسیده. اگر این ترن اسپل نرسیده باشد خروجی null است.

```
Void upgradeUnitRange(Unit)
```

```
Void upgradeUnitDamage(Unit)
```

به ترتیب رنج و دمیج یونیت داده شده را آپگرید میکند. در صورت معتبر بودن یکی از توکن های آپگرید کم می شود.

```
List[BaseUnit] getAllBaseUnits()
```

لیست تمام یونیت های پایه بازی را میدهد.

```
List[Spell] getAllSpells()
```

لیست تمام اسپل های بازی را میدهد.

```
King getKingById(playerId)
```

```
Spell getSpellById(spellId)
```

```
BaseUnit getBaseUnitById(typeId)
```

```
Player getPlayerById(playerId)
```

```
Unit getUnitById(unitId)
```

این توابع با ورودی گرفتن آیدی آبجکت ها خود آبجکت را باز می گردانند.

```
GameConstants getGameConstants()
```

ثابت های بازی را برمیگرداند.



AI CHALLENGE

2020

Player

| | |
|---|--|
| <code>int playerId</code> | آیدی بازیکن |
| <code>List[BaseUnit] deck</code> | دک بازیکن. تنها برای بازیکن خودی(و نه همتیمی و حریفها) معتبر است |
| <code>List[BaseUnit] hand</code> | دست بازیکن. تنها برای بازیکن خودی(و نه همتیمی و حریفها) معتبر است |
| <code>int ap</code> | میزان ایپی باقی مانده. تنها برای بازیکن خودی(و نه هم تیمی و حریفها) معتبر است |
| <code>King king</code> | کینگ بازیکن مسیرهایی که از کینگ بازیکن شروع میشوند. خانه ابتدای مسیر روی کینگ اوست. مسیر بین همتیمیها جزو آن نیست. |
| <code>List[Path] pathsFromPlayer</code> <code>Path pathToFriend</code> | مسیر بازیکن به هم تیمی اش.ابتدایش از آن کینگ شروع می شود. |
| <code>List[Unit] units</code> | یونیت‌های زنده بازیکن |
| <code>CastAreaSpell castAreaSpell</code> | اسپل محیطی که در ترن قبل توسط این بازیکن کست شده بود. اگر چیزی کست نشده باشد نال است. |
| <code>CastUnitSpell castUnitSpell</code> | اسپل یونیت که در ترن قبل توسط این بازیکن کست شده بود. اگر چیزی کست نشده باشد نال است. |
| <code>List[Unit] duplicateUnits</code> | یونیت های داپلیکیت و زنده بازیکن |
| <code>List[Unit] hastedUnits</code> | یونیت های هیست و زنده بازیکن |
| <code>List[Unit] playedUnits</code> | یونیت هایی که بازیکن آن ها را در ترن قبل بازی کرد (روی یک مسیر قرار داد) |
| <code>List[Unit] diedUnits</code> | یونیت های بازیکن که در ترن قبل مردند. |
| <code>Unit rangeUpgradedUnit</code> | یونیتی که ترن قبل توسط بازیکن رنجش آپگرید شد. اگر ترن قبل بازیکن آپگریدی روی رنج انجام نداده باشد، نال است. |



AI CHALLENGE

2020

| | |
|---------------------------------------|--|
| <code>Unit damageUpgradedUnit</code> | یونیتی که ترن قبل بازیکن دمیج آپگرید شد. اگر ترن قبل بازیکن آپگریدی روی دمیج انجام نداده باشد، نال است. |
| <code>List[Spell] spells</code> | اسپل های قابل استفاده بازیکن را میدهد. تنها برای بازیکن خودی و هم تیمی معتبر است. |
| <code>bool isAlive()</code> | نشان میدهد بازیکن زنده است یا نه که معادل با زنده بودن کینگ اوست. |
| <code>int getHp()</code> | سلامتی بازیکن را میدهد که معادل با سلامتی کینگ اوست. |
| <code>int getSpellCount(Spell)</code> | تعداد اسپل های اسپل داده شده که بازیکن میتواند استفاده کند را برمیگرداند. تنها برای بازیکن خودی و هم تیمی معتبر است. |

Unit

| | |
|--------------------------------|---|
| <code>BaseUnit baseUnit</code> | اطلاعات پایه یونیت در این آبجکت قرار دارد. |
| <code>Cell cell</code> | خانه ای که یونیت در آن قرار دارد. |
| <code>int unitId</code> | آیدی یونیت |
| <code>int hp</code> | سلامتی یونیت |
| <code>Path path</code> | مسیری که یونیت روی آن قرار دارد. این مسیر برای یونیت های دشمن نال است. |
| <code>Unit target</code> | کسی که قرار است این یونیت به آن حمله کند. اگر به کسی حمله نمی کند یا به کینگ حمله میکند مقدارش نال است |
| <code>Cell targetCell</code> | خانه ای که یونیت به آن حمله میکند. اگر به کینگ حمله میکند هم آن خانه از کینگ که به آن حمله میکند را برمیگرداند. |



AI CHALLENGE

2020

King targetIfKing

در صورتی که یونیت قرار است به کینگ حمله کند،
آن کینگ را برمیگرداند.

| | |
|-----------------|--|
| int playerId | آیدی بازیکنی که این یونیت برای اوست |
| int damageLevel | لول دمیج یونیت که با آپگرید زیاد می شود. در ابتدا ۰ است. |

| | |
|----------------|---|
| int rangeLevel | لول رنج یونیت که با آپگرید زیاد می شود. در ابتدا ۰ است. |
|----------------|---|

| | |
|-----------|-----------|
| int range | رنج یونیت |
|-----------|-----------|

| | |
|------------|------------|
| int attack | دمیج یونیت |
|------------|------------|

| | |
|------------------|-------------------------------------|
| bool isDuplicate | نشان میدهد یونیت داپلیکیت هست یا نه |
|------------------|-------------------------------------|

| | |
|---------------|--|
| bool isHasted | نشان میدهد یونیت هیست شده یا نه (سرعتش زیاد شده یا نه) |
|---------------|--|

| | |
|--------------------------------|--|
| List[CastSpell] affectedSpells | اسپل هایی که روی یونیت خورده بودند و ترن قبل تاثیر گذاشتند را برمیگرداند. |
|--------------------------------|--|

King

| | |
|-------------|------------------------|
| Cell center | خانه وسط کینگ را میدهد |
|-------------|------------------------|

| | |
|--------|-------------|
| int hp | سلامتی کینگ |
|--------|-------------|

| | |
|------------|-----------|
| int attack | دمیج کینگ |
|------------|-----------|

| | |
|-----------|---|
| int range | رنج کینگ که از خانه وسط آن محاسبه می شود. |
|-----------|---|

| | |
|--------------|-----------------|
| bool isAlive | زنده است یا خیر |
|--------------|-----------------|

| | |
|--------------|-----------------------|
| int playerId | آیدی بازیکن صاحب کینگ |
|--------------|-----------------------|

| | |
|-----------------|-----------------------------------|
| Cell targetCell | خانه ای که حمله کینگ آنجا میخورد. |
|-----------------|-----------------------------------|



AI CHALLENGE 2020

Map

| | |
|---|--|
| <code>int rowNum</code> | تعداد سطرهای مپ |
| <code>int colNum</code> | تعداد ستونهای مپ |
| <code>List[Path] paths</code> | لیست تمام مسیرهای مپ |
| <code>List[Unit] units</code> | لیست یونیت‌های زنده داخل مپ |
| <code>List[King] kings</code> | لیست کینگ‌های مپ |
| <code>Cell[][] cells</code> | لیست دو بعدی خانه های مپ |
| <code>Cell getCell(int row, int col)</code> | خانه با مختصات داده شده را برمیگرداند. |

Cell

| | |
|----------------------|------------|
| <code>int row</code> | شماره سطر |
| <code>int col</code> | شماره ستون |

Path

| | |
|-------------------------------|--------------------|
| <code>int id</code> | آیدی مسیر |
| <code>List[Cell] cells</code> | لیست خانه‌های مسیر |

BaseUnit

| | |
|------------------------------------|---|
| <code>int typeId</code> | آیدی یونیت پایه |
| <code>int maxHp</code> | سقف سلامتی یونیت |
| <code>int baseAttack</code> | دمیج اولیه یونیت |
| <code>int baseRange</code> | رنج اولیه یونیت |
| <code>UnitTarget targetType</code> | اینکه یونیت چه دسته ای از دشمنان را هدف قرار می‌دهد را نشان می‌دهد. |
| <code>bool isFlying</code> | اینکه یونیت روی هوا راه میرود یا روی زمین را نشان می‌دهد. |



AI CHALLENGE

2020

| | |
|------------------------------|--|
| <code>bool isMultiple</code> | اینکه یونیت به تمام یونیت های خانه حمله میکند یا تنها به یک نفر را نشان میدهد. |
| <code>int ap</code> | میزان ایپی لازم برای قرار دادن یونیت روی مپ |

Enum UnitTarget

| | |
|---------------------|--|
| <code>GROUND</code> | اینکه یونیت چه دسته ای از دشمنان را هدف قرار می دهد را نشان میدهد. |
| <code>AIR</code> | |
| <code>BOTH</code> | |

Spell

| | |
|----------------------------------|---|
| <code>SpellType type</code> | نوع اسپل را نشان میدهد (HP, TELE, HASTE,) (DUPLICATE) |
| <code>int typeId</code> | آیدی اسپل را نشان میدهد. این با فیلد قبل فرق دارد چون ممکن است از یک تایپ دو نوع اسپل وجود داشته باشد. |
| <code>int duration</code> | زمان اثر اسپل را نشان میدهد |
| <code>int priority</code> | اولویت اسپل را نشان میدهد. یعنی اسپل با اولویت کمتر زودتر اعمال می شود. |
| <code>SpellTarget target</code> | نوع هدف اسپل را نشان میدهد. |
| <code>int range</code> | رنج اسپل |
| <code>int power</code> | قدرت اسپل را نشان میدهد. در صورتی که اسپل دمیج وارد میکند مقدارش منفی است. |
| <code>bool isDamaging</code> | نشان میدهد که اسپل دمیج وارد میکند یا خیر که در صورتی که دمیج وارد کند حتما روی یونیت های دشمن اثر میگذارد. (target آن ENEMY میباشد). |
| <code>bool isAreaSpell()</code> | اگر اسپل محیطی باشد |
| <code>bool isUnitSpell()</code> | اگر اسپل یونیتی باشد |



AI CHALLENGE

2020

Enum SpellType

| | |
|-----------|----------|
| HP | نوع اسپل |
| TELE | |
| DUPLICATE | |
| HASTE | |

Enum SpellTarget

| | |
|--------|---|
| SELF | این یعنی اسپل تنها روی یونیت های بازیکن کست کننده اعمال می شود. |
| ALLIED | این یعنی اسپل روی یونیت های بازیکن کست کننده و هم تیمی اش اعمال می شود. |
| ENEMY | این یعنی اسپل روی یونیت های بازیکن های حریف اعمال می شود. |

CastSpell

| | |
|--------------------------|---|
| Spell spell | اسپلی که کست شد را نشان میدهد. |
| int id | آیدی کست اسپل |
| int casterId | آیدی بازیکنی که اسپل را کست کرد |
| Cell cell | برای اسپل AreaSpell خانه مرکز اسپل را نشان میدهد، برای اسپل یونیتی (تلیپورت) خانه ای که یونیت به آن رفته را نشان میدهد. |
| List[Unit] affectedUnits | لیست یونیت هایی که اسپل رویشان اثر گذاشته را میدهد. |



AI CHALLENGE

2020

CastAreaSpell

این کلاس از کلاس CastSpell ارث‌بری میکند و مربوط به اسپل های AreaSpell که کست شدند میباشد.

| | |
|---------------------|---|
| int reamainingTurns | تعداد ترن باقی‌مانده تا پایان تاثیر اسپل را نشان میدهد. |
|---------------------|---|

CastUnitSpell

| | |
|-----------|--|
| Unit unit | یونیتی که اسپل روی آن اعمال شده را نشان میدهد. |
|-----------|--|

| | |
|-----------|---|
| Path path | مسیری که یونیت روی آن رفته را نشان میدهد. اگر بازیکن کست کننده جزو دشمنان باشد، این متغیر نال می شود. |
|-----------|---|

GameConstants

| | |
|-----------|----------------------|
| int maxAp | سقف ap را نشان میدهد |
|-----------|----------------------|

| | |
|--------------|----------------|
| int maxTurns | تعداد ترن بازی |
|--------------|----------------|

| | |
|-----------------|---|
| int turnTimeout | زمان هر ترن را به میلی‌ثانیه نشان میدهد |
|-----------------|---|

| | |
|-----------------|--------------------------------------|
| int pickTimeout | زمان پیک را به میلی‌ثانیه نشان میدهد |
|-----------------|--------------------------------------|

| | |
|--------------------|---|
| int turnsToUpgrade | تعداد ترنی که یک توکن آپگرید به بازیکنان داده می شود. مثلا اگر ۲۳ باشد یعنی هر ۲۳ ترن یکبار یک توکن به هر بازیکن داده می شود. |
|--------------------|---|

| | |
|------------------|---|
| int turnsToSpell | تعداد ترنی که اسپل به بازیکنان داده می شود. مثلا اگر ۱۰ باشد یعنی هر ۱۰ ترن یکبار یک اسپل به هر بازیکن داده می شود. |
|------------------|---|

| | |
|---------------------------|--|
| int damageUpgradeAddition | با هر آپگرید رنج چه میزان رنج افزایش می‌یابد |
|---------------------------|--|

| | |
|--------------------------|---|
| int rangeUpgradeAddition | با هر آپگرید دمیج چه میزان دمیج افزایش مییابد |
|--------------------------|---|

| | |
|--------------|---------|
| int deckSize | سایز دک |
|--------------|---------|

| | |
|--------------|----------|
| int handSize | سایز دست |
|--------------|----------|



AI CHALLENGE 2020

راهنمای اپلیکیشن بازی



اجرای فایل سرور

بعد از زدن دکمه‌ی RUN SERVER وارد صفحه‌ی زیر می‌شویم. در این صفحه زمان تاخیر سرور را انتخاب می‌کنیم.



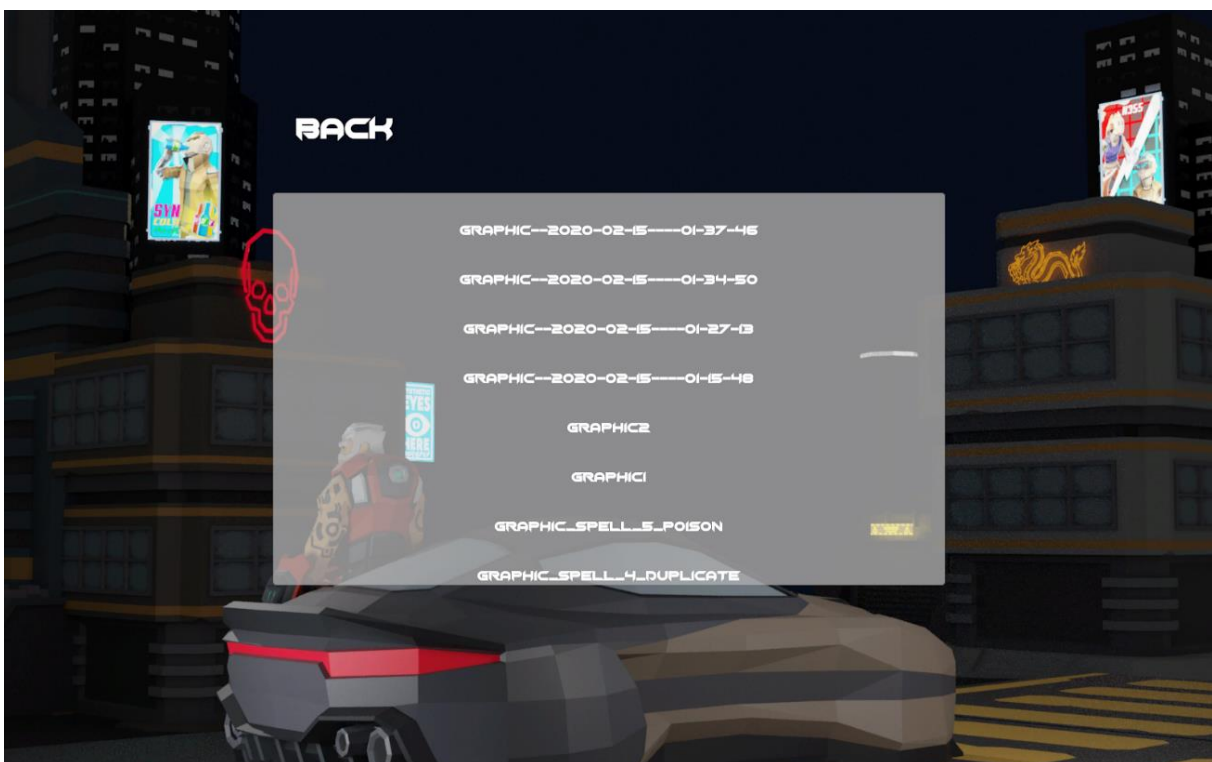


AI CHALLENGE 2020

با انتخاب زمان تاخیر فایل jar سرور اجرا می‌شود و باید فایل نقشه را در صفحه باز شده انتخاب کنید. برنامه سرور تا زمانی که کلاپنت‌ها اجرا نشده‌اند و بازی به اتمام نرسیده باز می‌ماند. بعد از اجرای بازی فایل لاگ بازی ساخته می‌شود.

اجرای لاگ بازی‌ها

بعد از زدن دکمه‌ی RUN LOG صفحه‌ی زیر باز می‌شود. در این صفحه لاگ مورد نظر برای اجرا را انتخاب کنید.





AI CHALLENGE 2020

نمایش گر بازی



برای حرکت دوربین در نمایش گر لاگ بازی می توانید از دکمه های زیر استفاده کنید.

E: حرکت به سمت جلو

Q: حرکت به سمت عقب

A: حرکت به سمت چپ

D: حرکت به سمت راست

W: حرکت به سمت مکانی که دوربین به آن سمت است

S: حرکت خلاف مکانی که دوربین به آن سمت است

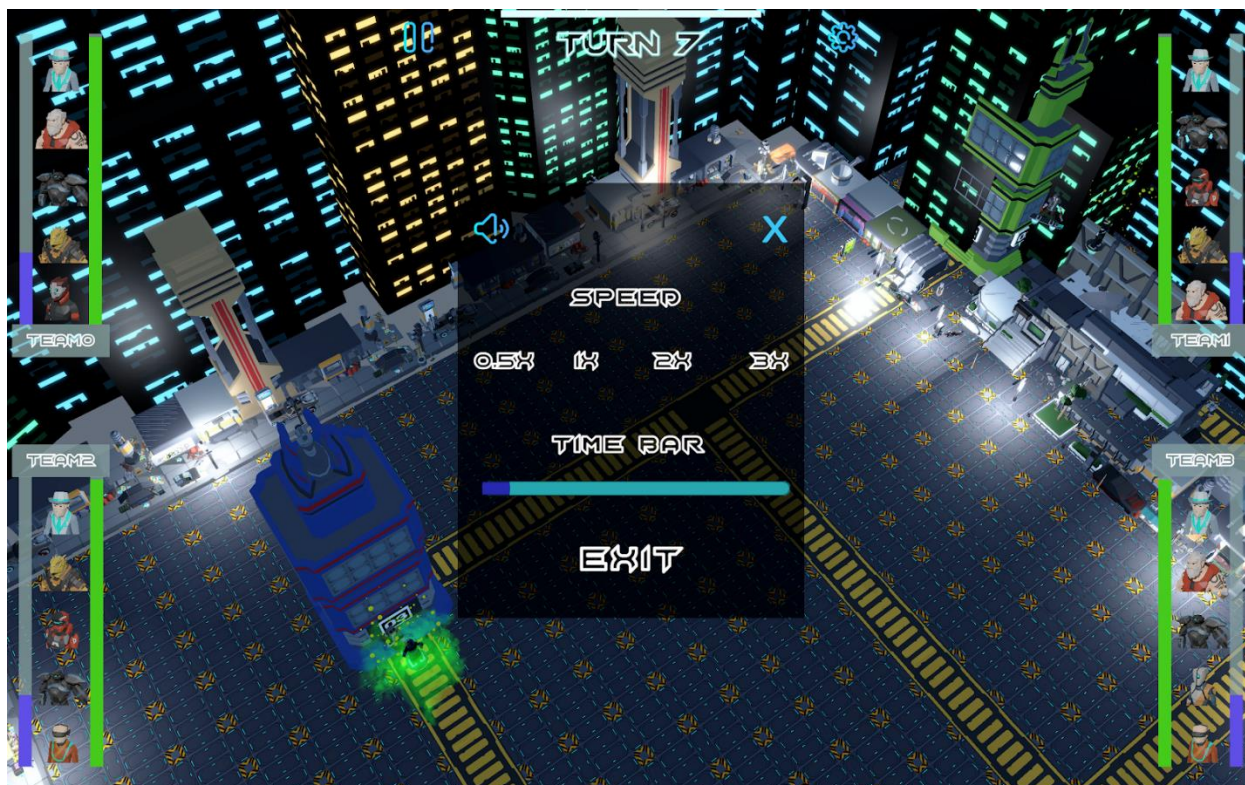
C: تغییر دوربین

همچنین برای تغییر جهت دوربین با نگهداشتن راست کلیک موس، موس را بچرخانید.



AI CHALLENGE 2020

برای تغییر سرعت اجرای بازی و تنظیمات دیگر از منوی تنظیمات که از طریق دکمه‌ی چرخ‌دنده‌ی بالای صفحه قابل دسترسی است، استفاده کنید.



محل قرارگیری فایل لاگ‌ها بعد از دانلود

برای دیدن لاگ‌های دانلود شده از سایت در اپلیکیشن با توجه به سیستم عامل خود از روش‌های زیر استفاده کنید.

مک: روی آیکون بازی راست کلیک کرده show package contents را انتخاب کنید سپس فایل لاگ را در فولدر Contents/Server/Log قرار دهید.

ویندوز و لینوکس: فایل لاگ را در فولدر AIC_Data/Server/Log قرار دهید. توجه کنید فولدر AIC_Data همیشه کنار فایل بازی باشد.