بسمه تعالی

مستند مسابقه

یازدهمین نبرد هوش مصنوعی شریف

مرحله‌ی غیر حضوری

بهمن و اسفند ۱۳۹۷

وضعیت بازی:

تیم‌ها: در هر مسابقه، تیم‌های شرکت کننده در قالب دو تیم با یک‌دیگر به رقابت می‌پردازند. هر تیم شامل ۴ قهرمان می‌باشد. ویژگی‌های تیم به شرح زیر است:

- **امتیاز عمل (AP):** هر تیم مشخصه‌ای تحت عنوان "امتیاز عمل" دارد که برای استفاده از قابلیت‌ها و حرکت قهرمانان خود، از این مشخصه هزینه می‌کند. در انتهای هر نوبت، مقدار این مشخصه به حداکثر خود بازمی‌گردد.
- **دید (Vision):** تمامی خانه‌های معمولی نقشه که در دید قهرمانان تیم قرار دارد، برای کل تیم قابل مشاهده است.

دو تیم روی یک نقشه به مبارزه می‌پردازند. نقشه‌ی بازی به شکل یک آرایه‌ی دو بعدی از خانه هاست. اولین درایه را سطر و دومین درایه را ستون می‌نامیم. خانه‌ی مبدا (۰،۰) نیز در بالا و چپ نقشه قرار دارد.

در هر لحظه اجزای زیر در نقشه‌ی بازی وجود دارد:

خانه های نقشه:

خانه های معمولی (Normal Cell): قهرمانان دو تیم می‌توانند به این خانه‌ها وارد شوند. با این محدودیت که در هر لحظه، حداکثر یک قهرمان از هر تیم در یک خانه‌ی مشخّص می‌تواند وجود داشته باشد. در یک خانه دو قهرمان از تیم‌های متفاوت به طور هم‌زمان می‌توانند حضور داشته باشند. مرزهای خانه‌ی معمولی جزء آن ***محسوب نمی‌شود***.

خانه های مسدود (Wall Cell): هیچ قهرمانی به این خانه‌ها نمی‌تواند وارد شود. هم‌چنین این خانه‌ها، دید قهرمانان و مسیر قابلیت‌های خطّی را مسدود می‌سازند. مرزهای خانه‌ی مسدود جزء آن ***محسوب می‌شود***.

منطقه های خاص:

منطقه‌ی هدف (Objective Zone): مجموعه‌ای از خانه‌های نقشه که حضور قهرمان در هر نوبت در آن، برای تیم امتیاز به همراه دارد.

منطقه‌ی زنده شدن (Respawn Zone): به ازای هر تیم یکی از این محدوده‌ها در نقشه وجود دارد. قهرمان‌ها پس از مرگ و گذشتن مدت مرگ، به تصادف در یکی از خانه‌های این محدوده زنده می‌شوند. این منطقه، منطقه‌ای امن است و نمی‌توان از داخل آن به خارج و برعکس آسیبی زد.

قهرمانان (Hero): قهرمانان در ۴ کلاس نگهبان، درمانگر، ویرانگر و قراول می‌باشند. هرکلاس از قهرمان‌ها، قابلیت‌های ویژه‌ی خود را دارد.

ویژگی‌های هر قهرمان به شرح زیر است:

دید (Vision):

یک قهرمان به یک خانه دید دارد اگر در میان خط واصل مرکز خانه‌ی کنونی و مرکز خانه‌ی مقصد، هیچ خانه‌ی مسدودی نباشد. برد دید تمامی قهرمانان بی‌نهایت است.

سلامتی اولیه (Max HP):

مقدار اولیه‌ی سلامتی قهرمان پس از زنده شدن.

سلامتی فعلی (Current HP):

میزان سلامت فعلی قهرمان. با رسیدن به صفر یا کمتر قهرمان می‌میرد.

مدت زمان زنده شدن (Respawn Time):

پس از مرگ قهرمان، با گذشتن این تعداد نوبت، قهرمان در خانه‌ای از منطقه‌ی زنده شدن تیم خودی به تصادف زنده می‌شود.

مدت زمان زنده شدن باقی‌مانده (Remaining Respawn Time) :

در صورت مرده بودن، قهرمان پس از این تعداد نوبت زنده خواهد شد.

هزینه‌ی حرکت (Move Cost):

به ازای هر خانه حرکت قهرمان، به این میزان از امتیاز عمل بازیکن کم می‌شود.

قابلیت (Ability):

هر قهرمان ۳ قابلیت دارد. مشخصات هر قابلیت به صورت زیر است:

- **نوع قابلیت (Ability Type):** نوع قابلیت به یکی از سه شکل زیر است و به همین ترتیب نیز اعمال می‌شوند:
  - **دفاعی (Defensive):** قابلیت هایی که بر قهرمانان خودی اثر می‌گذارد. مانند افزایش سلامتی.
  - **گریز (Dodge):** جابجایی سریع و آنی خود قهرمان در نقشه. با مشخص کردن یک خانه، قهرمان در همین نوبت به آن خانه جابجا می‌شود. از این قابلیت‌ها برای گریز از حمله‌ی دشمن می‌توان استفاده کرد.
  - **تهاجمی (Offensive):** مهارت هایی که بر قهرمانان حریف اثر می‌گذارد. مانند کاهش سلامتی.
- **هزینه‌ی انجام (Ability Cost):** با انجام قابلیت، این میزان از امتیاز عمل بازیکن کم می‌شود.
- **مدت زمان غیرفعال (Cooldown):** پس از استفاده، قابلیت پس از این تعداد نوبت قابل استفاده می‌شود.
- **قدرت (Power):** مقدار اثرگذاری قابلیت. برای قابلیت‌های تهاجمی میزان آسیب وارده به حریف و برای قابلیت‌های تدافعی میزان افزایش سلامتی قهرمان خودی می‌باشد.
- **برد اثرگذاری (Area Of Effect):** تا این فاصله از خانه‌ی مقصد را تحت تاثیر قرار می‌دهد.
- **حداکثر فاصله اعمال (Range):** حداکثر فاصله از قهرمان که می‌توان به عنوان خانه‌ی مقصد قابلیت انتخاب کرد.
- **پرتابی/خطّی (Is Lobbing) :** قابلیت‌های پرتابی مستقیماً در خانه‌ی مقصد اعمال می‌شوند و در واقع چیزی در میان مسیر نمی‌تواند جلوی آن‌را بگیرد. امّا قابلیت‌های خطّی، در صورت اصابت با دیوار یا هیرو در مسیر خود، همان‌جا اعمال اثر می‌کنند و به خانه‌ی مقصد نمی‌رسند.

روند بازی:

فاز پیش‌پردازش (PreProcess Turn): پیش از هرچیز، طی یک نوبت به هرتیم اجازه‌ی انجام پیش‌پردازش‌هایی داده می‌شود.

فاز انتخاب (Pick Turn): در ابتدای بازی، دو طرف به انتخاب قهرمان‌های خود می‌پردازند. این انتخاب در ۴ مرحله به صورت هم‌زمان صورت می‌گیرد و در پایان هر مرحله، قهرمانان انتخاب شده تا اینجا به دو طرف معرّفی می‌شوند.

پس از پایان فاز انتخاب، بازی آغاز می‌شود. هر نوبت بازی شامل ۶ فاز حرکت و ۱ فاز عمل است. پس از اتمام هر نوبت، امتیاز عمل هر تیم به مقدار اولیه باز می‌گردد.

فازهای حرکت (Move Turn): در هرکدام از این فازها، هر قهرمان را می‌توان حداکثر به یکی از خانه های مجاور ضلعی جابه‌جا کرد. پس از تصمیم گیری دو طرف و در انتهای هر فاز، حرکات به شکل هم‌زمان صورت می‌گیرند.

فاز عمل (Action Turn): در این فاز، حداکثر یک قابلیت از هر قهرمان را می‌توان استفاده کرد. در انتهای فاز، به ترتیب نوع قابلیت‌ها، یعنی ابتدا دفاعی‌ها، سپس گریزها و در انتها تهاجمی‌ها اعمال می‌شوند.

در انتهای هر نوبت، به ازای تعداد قهرمانان زنده‌ی موجود در منطقه‌ی هدف، و ۱۰ برابر تعداد قهرمانان کشته شده از حریف در این نوبت، به امتیاز تیم مربوطه اضافه می‌شود.

پایان بازی:

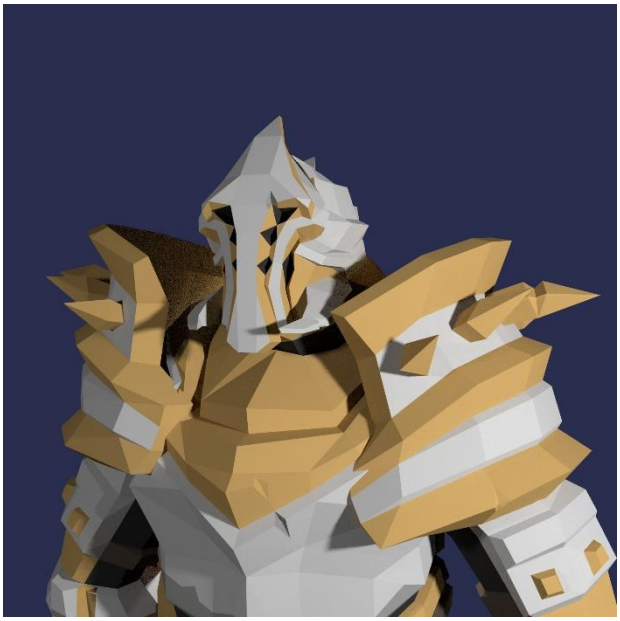
اولین تیمی که به ۲۰۰ امتیاز برسد پیروز خواهد شد. در غیر این‌صورت بازی به مدت ۱۰۰ نوبت انجام می‌پذیرد. در انتها تیمی که امتیاز بیشتری کسب کرده باشد، پیروز خواهد شد. امتیاز هر تیم شامل مجموع امتیاز کشتن قهرمانان حریف و مجموع امتیاز حضور در محدوده‌ی هدف می‌باشد. در صورت تساوی امتیازات، تیمی که در مجموع ۱۰۰ نوبت امتیاز عمل کمتری استفاده کرده، پیروز خواهد شد. در صورت تساوی در مراحل انتخابی و حضوری، تیمی که کد خود را زودتر ارسال کرده، برنده اعلام می‌شود.

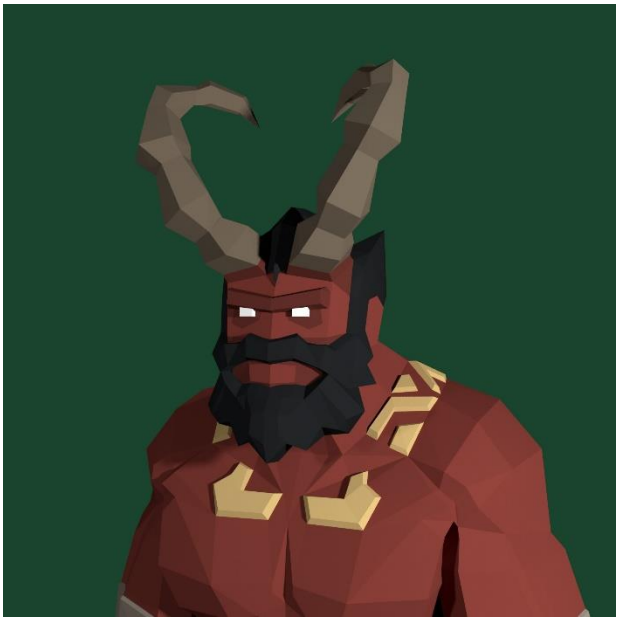
ویژگی‌های بازی:

نام ویژگی	مقدار ویژگی
امتیاز عمل اولیه (AP)	۱۰۰
حداکثر امتیاز پیروزی (Max Score)	۲۰۰
حداکثر نوبت بازی (Max Turn)	۱۰۰
تعداد فازهای انتخاب (Pick Turn)	۴
تعداد فازهای حرکت در هر نوبت (Move Turn)	۶
تعداد فازهای عمل در هر نوبت (Action Turn)	۱
امتیاز کشتن یک قهرمان از حریف (Kill Score)	۱۰
امتیاز منطقه‌ی هدف (Objective Zone Score)	۱

در ادامه به معرفی کلاس‌های مختلف قهرمان‌ها می‌پردازیم.

ویرانگر





توضیحات: ویرانگر نسبت به دیگر قهرمان ها از سرعت خوبی برخوردار است. قابلیت خاص این قهرمان، پرتاب بمب با قدرت آسیب و محدوده اثرگذاری بالاست که می‌توان چند قهرمان کنار هم را در یک نوبت هدف قرار داد.

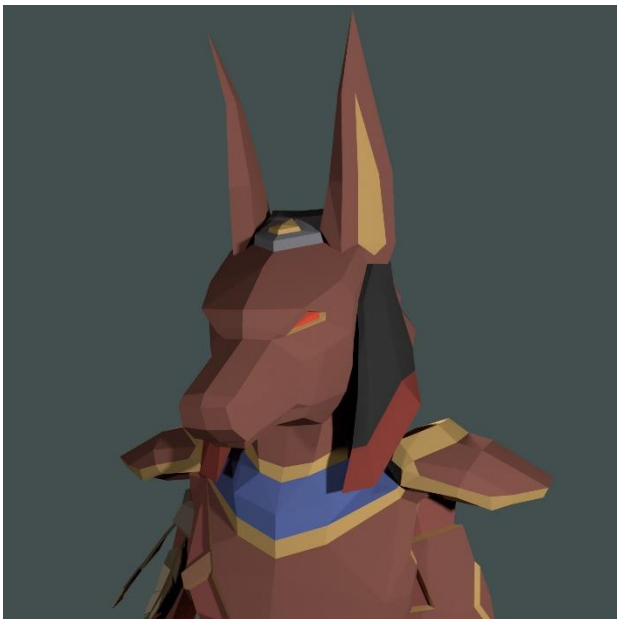
ویژگی های کلی:

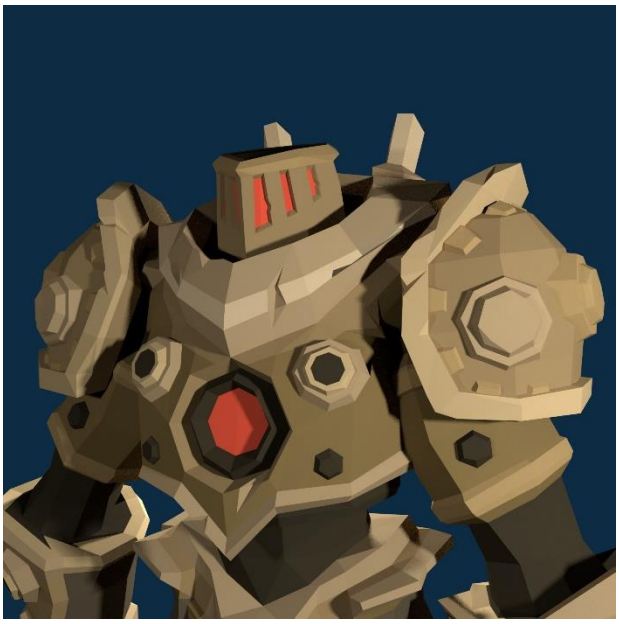
نام ویژگی	مقدار ویژگی
سلامتی اولیه	۲۵۰
مدت زمان زنده شدن	۵
هزینه‌ی حرکت	۴

ویژگی های قابلیت‌ها:

نام ویژگی	حمله	گریز	بمب
نوع قابلیت	تهاجمی	گریز	تهاجمی
هزینه‌ی انجام	۱۵	۲۵	۲۵
مدت زمان غیرفعال	۱	۶	۴
قدرت	۳۰	-	۴۰
برد اثرگذاری	۱	-	۲
حداکثر فاصله‌ی اعمال	۴	۴	۵
پرتابی/خطّی	خطّی	پرتابی	پرتابی

قراول





توضیحات: قراول قهرمانی دوربرد اما نسبتا کند است. قابلیت خاص این قهرمان، شلیک یک اشعه با برد نامحدود و قدرت آسیب بالاست.

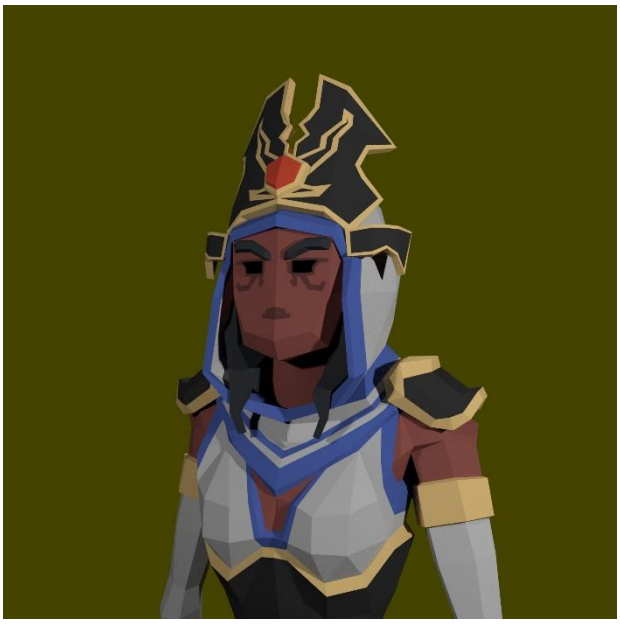
ویژگی های کلی:

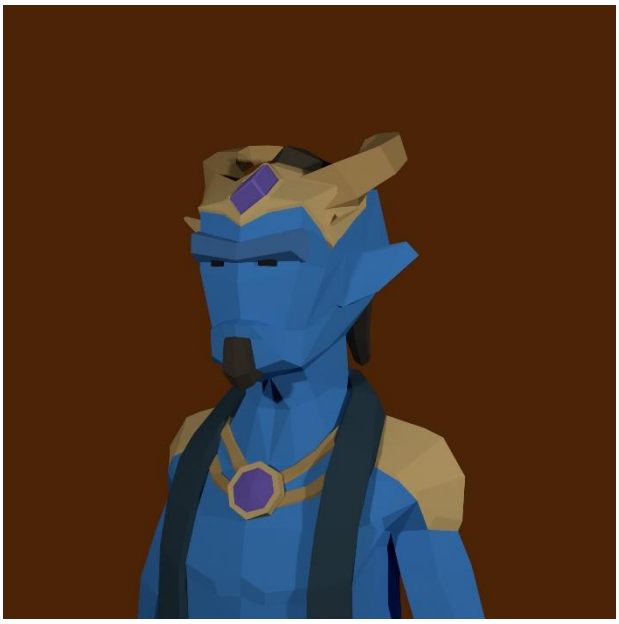
نام ویژگی	مقدار ویژگی
سلامتی اولیه	۱۲۰
مدت زمان زنده شدن	۵
هزینه‌ی حرکت	۶

ویژگی های قابلیت‌ها:

نام ویژگی	حمله	گریز	اشعه
نوع قابلیت	تهاجمی	گریز	تهاجمی
هزینه‌ی انجام	۱۵	۲۵	۲۵
مدت زمان غیرفعال	۱	۷	۵
قدرت	۳۰	-	۵۰
برد اثرگذاری	۰	-	۰
حداکثر فاصله‌ی اعمال	۷	۳	نامحدود
پرتابی/خطّی	خطّی	پرتابی	خطّی

درمانگر





توضیحات: درمانگر قهرمانی با سرعت و سلامتی متوسط است. قابلیت خاص این قهرمان، درمانگری است که می‌تواند سلامتی یک قهرمان خودی را حداکثر ۴۰ واحد تا مقدار سلامتی اولیه افزایش دهد.

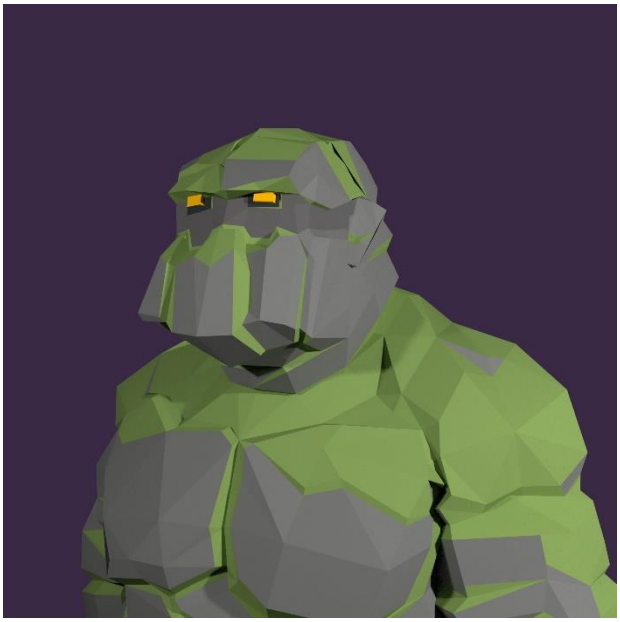
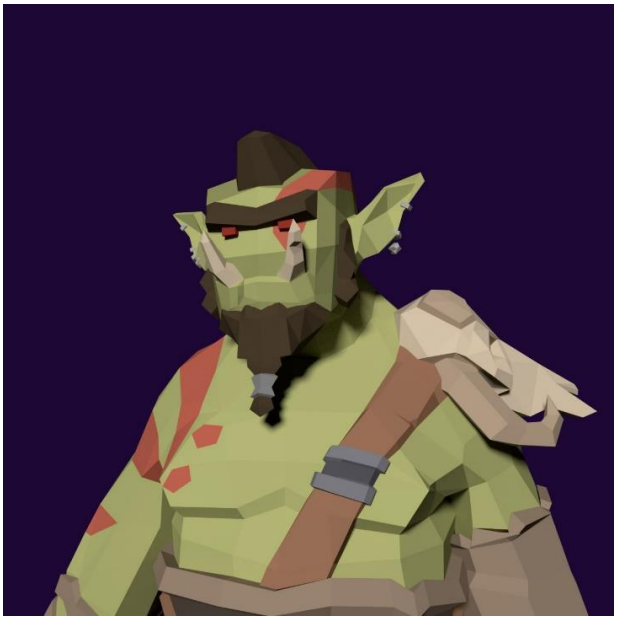
ویژگی های کلی:

نام ویژگی	مقدار ویژگی
سلامتی اولیه	۲۰۰
مدت زمان زنده شدن	۵
هزینه‌ی حرکت	۴

ویژگی های قابلیت‌ها:

نام ویژگی	حمله	گریز	درمانگری
نوع قابلیت	تهاجمی	گریز	تدافعی
هزینه‌ی انجام	۱۵	۲۵	۲۵
مدت زمان غیرفعال	۱	۶	۳
قدرت	۱۰	-	۴۰
برد اثرگذاری	۰	-	۰
حداکثر فاصله‌ی اعمال	۴	۴	۴
پرتابی/خطّی	پرتابی	پرتابی	پرتابی

نگهبان

توضیحات: نگهبان قهرمانی با سلامتی اولیه زیاد اما سرعتی بسیار کند است. قابلیت خاص این قهرمان، تقویت یک قهرمان خودی است که باعث می‌شود در این نوبت به مقدار نامحدود آسیب های وارده به قهرمان هدف بی اثر شود.

ویژگی های کلی:

نام ویژگی	مقدار ویژگی
سلامتی اولیه	۴۰۰
مدت زمان زنده شدن	۵
هزینه‌ی حرکت	۸

ویژگی های قابلیت‌ها:

نام ویژگی	حمله	گریز	تقویت
نوع قابلیت	تهاجمی	گریز	تدافعی
هزینه‌ی انجام	۱۵	۲۵	۲۵
مدت زمان غیرفعال	۱	۸	۸
قدرت	۲۰	-	نامحدود
برد اثرگذاری	۱	-	۰
حداکثر فاصله‌ی اعمال	۱	۲	۴
پرتابی/خطّی	خطّی	پرتابی	پرتابی

شروع کد نویسی (بایدها و نبایدها)

۱.شما باید کد هوش مصنوعی خود را در توابع preProcessTurn و pickTurn و moveTurn و actionTurn 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 (فایل‌هایی غیر از فایلی که در آن کد می‌زنید) را در نر بگیرید.

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

https://github.com/SharifAIChallenge/

(repo کلاینت‌ها با پسوند AIC19-Client آغاز می‌شود)

نحوه‌ی اجرا و توسعه (کلاینت جاوا)

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

نحوه‌ی 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 کافیست پوشه‌ای 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 پوشه‌ای 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 پوشه‌ی جاری ساخته خواهد شد.

نحوه‌ی اجرا و توسعه (کلاینت پایتون)

برای اجرای کلاینت پایتون نیاز به پایتون ۳ دارید. پس از نصب پایتون ۳ کافیست فایل Controller.py را اجرا نمایید. شما می‌توانید به این کلاینت فایل‌های مورد نظرتان را اضافه کنید یا فایل‌های آن را تغییر دهید به شرط آن که کد شما با سرور سازگار باشد . به سادگی بتوانید نسخه‌های احتمالی بعدی کلاینت را جایگزین نمایید. برای توسعه و کدنویسی به این زبان، PyCharm را توصیه می‌کنیم.

ارسال فایل ها

برای ارسال فایل‌های خود، پوشه‌ی کلاینت را فشرده کنید و فایل فشرده را ارسال نمایید.

رابط برنامه نویسی:

نکات مهم:

۱.رفرنس‌های موجود در نقشه، در هر نوبت از ابتدا ساخته می‌شوند و رفرنس‌های نوبت قبل معتبر نیستند. (برای حفظ دسترسی به آبجکت‌ها، توصیه می‌شود از id آن‌ها، که همواره ثابت است، استفاده کنید.)

۲. در تمامی قسمت هایی که علامت ستاره (*) وجود دارد، تمامی حالات تابع شامل شناسه‌ی قهرمان به جای آبجکت قهرمان، مختصات سطر و ستون به جای آبجکت خانه و نام قابلیت به جای آبجکت قابلیت پیاده شده است.

۳. در کلاینت پایتون، متدهای get پیاده نشده‌اند و به متغیرها دسترسی مستقیم وجود دارد.

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

Class (or interface) Ability	
`int getRemCooldown();`	مقدار باقی‌مانده از مدت زمان غیر فعال. این مقدار برای دشمن نامعتبر است.
`boolean isReady();`	در صورتی که قابلیت آماده‌ی انجام در این نوبت باشد، true برمی‌گرداند.
همه‌ی متدهای کلاس AbilityConstants در این کلاس نیز موجود است.	

Class (or interface) AbilityConstants	
`AbilityName getName();`	نام قابلیت
`AbilityType getType();`	نوع قابلیت
`int getRange();`	حداکثر فاصله‌ی اعمال
`int getAPCost();`	هزینه‌ی انجام قابلیت
`int getCooldown();`	مدت زمان غیر فعال بودن قابلیت پس از استفاده
`int getAreaOfEffect();`	برد اثرگذاری قابلیت
`int getPower();`	قدرت قابلیت
`boolean isLobbing();`	پرتابی/خطّی بودن قابلیت. برای پرتابی مقدار true و برای خطّی مقدار false.

enum AbilityName	
SENTRY_ATTACK	حمله‌ی قراول
SENTRY_DODGE	گریز قراول
SENTRY_RAY	اشعه‌ی قراول
BLASTER_ATTACK	حمله‌ی ویرانگر
BLASTER_DODGE	گریز ویرانگر
BLASTER_BOMB	بمب ویرانگر
HEALER_ATTACK	حمله‌ی درمانگر
HEALER_DODGE	گریز درمانگر
HEALER_HEAL	درمانگری درمانگر
GUARDIAN_ATTACK	حمله‌ی محافظ
GUARDIAN_DODGE	گریز محافظ
GUARDIAN_FORTIFY	تقویت محافظ

enum AbilityType	
DEFENSIVE	دفاعی
DODGE	گریز
OFFENSIVE	تهاجمی

Class (or interface) CastAbility	
int getCasterId();	شناسه‌ی قهرمان اعمال کننده‌ی قابلیت. در صورت عدم مشاهده‌ی قهرمان اعمال کننده، نامعتبر است.
int[] getTargetHeroIds();	شناسه‌های قهرمان‌های تحت تاثیر قابلیت که در دید هستند.
Cell getStartCell();	خانه‌ی مبدا اعمال قابلیت. در صورت عدم حضور این خانه در دید، نامعتبر است.
Cell getEndCell();	خانه‌ی مقصد اعمال قابلیت. در صورت عدم حضور این خانه در دید، نامعتبر است.
AbilityName getAbilityName();	نام قابلیت. در صورت نامعتبر بودن همه اطلاعات بالا، نامعتبر است.

Class (or interface) Cell	
int getRow();	شماره‌ی سطر خانه
int getColumn();	شماره‌ی ستون خانه
boolean isWall();	در صورت مسدود بودن خانه true و در صورت معمولی بودن false برمی‌گرداند.

`boolean isInMyRespawnZone();`	اگر در منطقه‌ی زنده شدن خودی باشد، true برمی‌گرداند.
`boolean isInOppRespawnZone();`	اگر در منطقه‌ی زنده شدن دشمن باشد، true برمی‌گرداند.
`boolean isInObjectiveZone();`	اگر در منطقه‌ی هدف باشد، true برمی‌گرداند.
`boolean isInVision();`	اگر در محدوده‌ی دید فعلی باشد، true برمی‌گرداند.

`enum Direction`	
`UP`	بالا
`DOWN`	پایین
`LEFT`	چپ
`RIGHT`	راست

`Class (or interface) Hero`	
`Ability getAbility(AbilityName);`	قابلیت با این نام را در صورت وجود برمی‌گرداند.
`int getId();`	شناسه‌ی قهرمان را برمی‌گرداند.
`Ability[] getAbilities();`	قابلیت‌های قهرمان را برمی‌گراند.
`Ability[] getDefensiveAbilities();`	قابلیت‌های دفاعی قهرمان را برمی‌گرداند.
`Ability[] getDodgeAbilities();`	قابلیت‌های گریز قهرمان را برمی‌گرداند.
`Ability[] getOffensiveAbilities();`	قابلیت‌های تهاجمی قهرمان را برمی‌گرداند.
`int getCurrentHP();`	سلامتی کنونی قهرمان. برای قهرمانان دشمن، نامعتبر است.
`Cell getCurrentCell();`	خانه‌ی کنونی قهرمان. برای قهرمانان دشمن خارج از دید، نامعتبر است.
`Cell[] getRecentPath();`	مسیر حرکت قهرمان از ابتدای نوبت جاری تاکنون. برای قهرمانان دشمن، حرکت هایی که در محدوده‌ی دید بوده برمی‌گردد.
`int getRemRespawnTime();`	مقدار باقی‌مانده از مدت زمان زنده شدن.
همه‌ی متدهای کلاس `HeroConstants` در این کلاس نیز موجود است.	

`Class (or interface) HeroConstants`	
`HeroName getName();`	نوع قهرمان را برمی‌گرداند.
`AbilityName[] getAbilityNames();`	نام قابلیت‌های قهرمان را برمی‌گرداند.
`int getMaxHP();`	سلامتی اولیه قهرمان
`int getMoveAPCost();`	هزینه حرکت قهرمان

`int getRespawnTime();`	مدت زمان زنده شدن

enum HeroName	
`SENTRY`	قراول
`BLASTER`	ویرانگر
`HEALER`	درمانگر
`GUARDIAN`	محافظ

enum Phase	
`PICK`	فاز انتخاب
`MOVE`	فاز حرکت
`ACTION`	فاز عمل

Class (or interface) Map	
`Cell[][] getCells();`	آرایه‌ی دوبعدی خانه‌های نقشه را برمی‌گرداند.
`Cell getCell(int, int);`	خانه‌ی مورد نظر در مختصات داده شده را برمی‌گرداند.
`boolean isInMap(int, int);`	در صورت خارج محدوده بودن مختصات، false برمی‌گرداند.
`int getRowNum();`	تعداد سطرهای نقشه
`int getColumnNum();`	تعداد ستون‌های نقشه
`Cell[] getMyRespawnZone();`	خانه‌های محدوده‌ی زنده شدن خودی را برمی‌گرداند.
`Cell[] getOppRespawnZone();`	خانه‌های محدوده‌ی زنده شدن دشمن را برمی‌گرداند.
`Cell[] getObjectiveZone();`	خانه‌های محدوده‌ی هدف را برمی‌گرداند.

Class (or interface) World	
`Hero getHero(int);`	قهرمان با شناسه‌ی داده شده را برمی‌گرداند.
`Hero getMyHero(Cell);` `Hero getMyHero(int, int);`	قهرمان خودی در خانه‌ی داده شده (مختصات) را برمی‌گرداند.
`Hero getOppHero(Cell);` `Hero getOppHero(int, int);`	قهرمان دشمن را در خانه‌ی داده شده (مختصات) در صورتی که در دید باشد، برمی‌گرداند.
`void castAbility(Hero, Ability, Cell);`	اعمال قابلیت قهرمان به خانه‌ی موردنظر.*
`void moveHero(Hero, Direction);` `void moveHero(int, Direction);`	حرکت قهرمان در جهت مورد نظر به اندازه‌ی یک خانه.
`void pickHero(HeroName);`	انتخاب یکی از این نوع قهرمان در فاز انتخاب.

`Direction[] getPathMoveDirections(Cell, Cell);`	دنباله‌ی جهت ها برای رسیدن از خانه‌ی اول به خانه‌ی دوم.*
`int manhattanDistance(Cell, Cell);`	فاصله منهتنی دو خانه. پیاده‌سازی با مختصات نیز موجود است.*
`Cell getImpactCell(Ability, Cell, Cell);`	در صورت استفاده از قابلیت خطّی، با استفاده از این تابع می‌توان خانه‌ی محل اثر را بر اساس دید فعلی بدست آورد.*
`Hero[] getAbilityTargets(Ability, Cell, Cell);`	در صورت اعمال قابلیت از خانه‌ی اول به دوم، قهرمانانی که مورد اثر قرار می‌گیرند و در دید هستند را برمی‌گرداند.*
`boolean isInVision(Cell, Cell);`	در صورت قابلیت دیدن خانه‌ی دوم از خانه‌ی اول، true برمی‌گرداند.*
`Hero[] getMyHeroes();`	قهرمانان خودی را برمی‌گرداند.
`Hero[] getOppHeroes();`	قهرمانان دشمن را برمی‌گرداند.
`Hero[] getMyDeadHeroes();`	قهرمانان مرده‌ی خودی را برمی‌گرداند.
`Hero[] getOppDeadHeroes();`	قهرمانان مرده‌ی دشمن را برمی‌گرداند.
`Map getMap();`	نقشه‌ی بازی را برمی‌گرداند.
`CastAbility[] getMyCastAbilities();`	قابلیت های اعمال شده‌ی خودی در نوبت قبلی.
`CastAbility[] getOppCastAbilities();`	قابلیت های اعمال و مشاهده شده‌ی دشمن در نوبت قبلی.
`HeroConstants[] getHeroConstants();`	ویژگی‌های انواع مختلف قهرمانان را برمی‌گرداند.
`AbilityConstants[] getAbilityConstants();`	ویژگی‌های انواع مختلف قابلیت‌ها را برمی‌گرداند.
`int getAP();`	مقدار امتیاز عمل فعلی را برمی‌گرداند.
`int getMyScore();`	امتیاز فعلی خودی را برمی‌گرداند.
`int getOppScore();`	امتیاز فعلی دشمن را برمی‌گرداند.
`int getCurrentTurn();`	شماره نوبت فعلی را برمی‌گرداند.
`Phase getCurrentPhase();`	فاز فعلی را برمی‌گرداند.
`int getMaxScore();`	مقدار سقف امتیاز بازی برای پیروزی را برمی‌گرداند.
`int getMaxAP();`	مقدار اولیه‌ی امتیاز عمل را برمی‌گرداند.
`int getMaxTurns();`	تعداد کل نوبت‌های بازی را برمی‌گرداند.
`int getKillScore();`	امتیاز حاصل از کشتن یک قهرمان دشمن را برمی‌گرداند.
`int getObjectiveZoneScore();`	امتیاز حاصل از حضور یک قهرمان خودی در منطقه‌ی هدف را برمی‌گرداند.

`int getMovePhaseNum();`	شماره‌ی فاز حرکت فعلی در نوبت جاری را برمی‌گرداند.

Class (or interface) AI	
`void preProcess(World);`	این تابع فقط در ابتدای شروع بازی و پیش از فاز انتخاب اول صدا زده می‌شود. محدودیت زمانی ۵ ثانیه
`void pickTurn(World);`	تابع فازهای انتخاب. به ازای هرکدام از این فازها، یکبار صدا زده می‌شود. محدودیت زمانی ۰٫۲ ثانیه.
`void moveTurn(World);`	تابع فازهای حرکت. به ازای هرکدام از این فازها، یکبار صدا زده می‌شود. محدودیت زمانی اولین فاز هر نوبت، ۰٫۵ ثانیه و بقیه ۰٫۲ ثانیه
`void actionTurn(World);`	تابع فازهای عمل. به ازای هرکدام از این فازها، یکبار صدا زده می‌شود. محدودیت زمانی ۰٫۲ ثانیه.