



**python/JavaScript client**

# AI Cup

2      0      2      3

کوش مصنوعی را همفاوت تجربه کنید!

## شروع و اجرای بازی

- برای اجرای کلاینت کافی است فایل `run.py` را اجرا کنید.
- استراتژی خود را باید در دو تابع `initialize` و `turn` که در فایل `main.py` قرار دارد بنویسید.

تابع `initialize` در بازی ۳۵ بار صدا زده میشود و هر بار که صدا زده میشود شما باید یک نیرو در یکی از خانه ها قرار دهید

تابع `turn` حداقل ۲۰ بار برای شما صدا زده میشود و شما در آن باید ۴ استیت نوبت را جلو بیرید و استراتژی خود را پیاده سازی کنید. در ورودی این تابع یک آبجکت از کلاس `Game` به شما داده میشود که در آن توابع مورد نیاز برای پیاده سازی استراتژی قرار دارد.

- برای مشاهده نمونه خروجی هر تابع روی آن کلیک کنید

## توابع بازی

### get\_owners()

- این تابع یک دیکشنری برمی‌گرداند که کلید های آن، شماره هر سیاره است و هر کلید مقداری برابر با شماره صاحب آن سیاره دارد . اگر سیاره‌ای صاحب نداشت مقدار آن برابر -1 است.

## get\_number\_of\_troops\_to\_put()

- این تابع یک دیکشنری برمی‌گرداند که کلید آن برابر با number\_of\_troops است و مقدار آن برابر با تعداد سربازی است که آن بازیکن در اختیار دارد ولی هنوز از آن استفاده نکرده است.

## get\_state()

- این تابع یک دیکشنری برمی‌گرداند که کلید آن برابر با state است و مقدار آن مشخص می‌کند که بازیکن چه عملیاتی می‌تواند انجام دهد.
- اگر مقدار آن برابر ۱ باشد یعنی بازیکن توانایی این را دارد که سرباز های خود را در یک سیاره مستقر کند. (استقرار سرباز)
- اگر مقدار آن برابر با ۲ باشد یعنی می‌تواند به سیاره‌ای حمله کند.
- اگر مقدار آن برابر ۳ باشد می‌تواند نیرو های خود را به سیاره ای دیگر منتقل کند. (جابجایی نیرو)
- اگر برابر با ۴ باشد یعنی در مرحله تبدیل نیرو های دفاعی هستید

### get\_adj()

- این تابع یک دیکشنری برمی‌گرداند که کلیدهای آن برابر با شماره های سیاره‌ها است و هر کلید مقداری برابر با یک لیست دارد که این لیست شامل شماره تمام همسایه‌های این سیاره است.

### get\_player\_id()

- این تابع یک دیکشنری برمی‌گرداند که کلید آن player\_id است و مقدار آن برابر با شماره بازیکن شما است.

### get\_strategic\_nodes()

- این تابع یک دیکشنری برمی‌گرداند که شامل ۲ تا کلید است :
- کلید strategic\_nodes : مقدار آن لیست شش سیاره‌ای است که به عنوان سیاره استراتژیک در بازی معرفی شده‌اند.
- کلید score : مقدار آن لیستی از امتیاز نظیر هر کدام از سیاره‌های استراتژیک است.

### **get\_number\_of\_troops()**

- این تابع یک دیکشنری برمی‌گرداند که کلیدهای آن شماره سیاره‌ها است و مقدار هر کلید برابر است با تعداد نیروهایی که در آن سیاره وجود دارد.

### **get\_reachable()**

- این تابع به عنوان ورودی شماره یک سیاره می‌گیرد و به عنوان خروجی یک دیکشنری برمی‌گرداند که کلید آن reachable است و مقدار آن برابر با لیستی از سیاره‌هایی است که بازیکن می‌تواند از سیاره‌ای که انتخاب کرده، به آن‌ها نیرو منتقل کند.

### **get\_number\_of\_fort\_troops()**

- این تابع یک دیکشنری برمی‌گرداند که کلیدهای آن شماره سیاره‌ها است و مقدار هر کدام از آن برابر با تعداد نیروهای دفاعی موجود در آن سیاره است.

### put\_one\_troop()

- این تابع به عنوان ورودی شماره سیاره‌ای را می‌گیرد که بازیکن در فاز شروع بازی قصد دارد درون آن یک نیرو قرار بدهد و در صورت موفقیت آمیز بودن، به عنوان خروجی یک دیکشنری می‌دهد که کلید آن برابر با message است و مقدار آن برابر با troop added است. message successfully
- در صورتی که این کار امکان پذیر نباشد برنامه برای شما یک خطای rase می‌کند که متن آن دلیل عدم موفقیت رو توضیح میدهد

### put\_troop()

- این تابع برای گذاشتن نیرو در استیت اول هر نوبت استفاده می‌شود و با استفاده از آن مشخص می‌کنید که در کدام خانه چه تعداد نیرو قرار بگیرد
- این تابع را تا زمانی که در استیت اول نوبت باشید می‌توانید صدا بزنید و نیروهایی که درون نقشه نگذاشته اید را درون نقشه بگذارید.
- این تابع ۲ ورودی می‌گیرد:
  - شماره سیاره‌ای که قصد دارد درون آن نیرو قرار دهد: node\_id
  - تعداد سربازی که قصد دارد روی آن سیاره قرار دهد: num

## • خروجی:

- در صورت موفقیت آمیز بودن یک دیکشنری به عنوان خروجی می‌دهد که کلید آن برابر با message است و مقدار آن برابر با troop added successfully است.
- در صورتی که این کار امکان پذیر نباشد برنامه برای شما یک خطای rase می‌کند که متن آن دلیل عدم موفقیت را توضیح میدهد

## attack()

- این تابع برای حمله به دیگر سیاره‌ها پس از مرحله قرار گیری نیروها استفاده می‌شود و ۴ تا ورودی دریافت می‌کند:
  - شماره سیاره ای که قصد داریم حمله از آن سیاره آغاز شود:  $\beta$  attacking\_id
  - شماره سیاره ای که قصد داریم به آن حمله کنیم:  $\beta$  target
  - حمله تا زمانی که نسبت نیروهای مهاجم به نیروهای مدافعانقل چه مقداری است، ادامه داشته باشد:  $\beta$  fraction
  - در صورت موفقیت آمیز بودن حمله، چه کسری از نیروهای باقی‌مانده از حمله به سیاره تصاحب شده انتقال یابند:  $\beta$  move\_fraction

**• خروجی:**

- در صورت موفقیت آمیز بودن یک دیکشنری به عنوان خروجی می‌دهد که کلید آن برابر با message است و مقدار آن برابر با attack is successful است.

**• شرایط حمله:**

- شماره سیاره ای که میخواهیم از آن حمله کنیم حتماً فقط باید شامل ارقام باشد (۰-۹)
- شماره سیاره‌ای که می‌خواهیم از آن حمله کنیم در بین سیاره‌های بازی باید
- سیاره‌ای که می‌خواهیم از آن حمله کنید متعلق به خودتان باشد
- سیاره‌ای که می‌خواهیم به آن حمله کنید متعلق به شما نباشد
- میزان نیروهای سیاره‌ای که می‌خواهیم از آن حمله کنیم از ۲ کمتر نباشد
- شماره سیاره‌ای که می‌خواهیم به آن حمله کنیم، حتماً باید فقط شامل ارقام باشد (۰-۹)
- به سیاراتی که متعلق به هیچ بازیکنی نیستند نمی‌توانید حمله کنید
- مقدار fraction داده شده باید قابلیت تبدیل شدن به عدد اعشاری را داشته باشد
- مقدار move\_fraction داده شده باید قابلیت تبدیل شدن به عدد اعشاری را داشته باشد
- مقدار move\_fraction باید مقداری بین ۰-۱ داشته باشد

## move\_troop()

- این تابع برای انتقال نیروها پس از مرحله حمله استفاده می‌شود و ۳ ورودی دریافت می‌کند:

- نیروهای کدام سیاره باید انتقال بیابند (مبدا):  $\beta$  source
- نیروهای سیاره انتخاب شده به کدام سیاره باید منتقل شوند  $\beta$  destination (مقصد)
- چه هیزان نیرو باید انتقال داده شود (تعداد نیروها):  $\beta$  troop\_count

### • خروجی:

- در صورت موفقیت آمیز بودن یک دیکشنری به عنوان خروجی می‌دهد که کلید آن برابر با message است و مقدار آن برابر با troops moved successfully است.

### • شرایط:

1. باید در فاز اصلی و استیت سه نوبت باشیم
2. باید سیاره‌ی مبدأ و مقصد متعلق به خودمان باشد
3. باید مسیری بین این دو سیاره وجود داشته باشد که تمام سیارات آن متعلق به خودمان باشد

## next\_state()

- زمانی که در نوبت خود می خواهید به استیت بعدی بروید باید این تابع را صدا بزنید مثلا در استیت گذاشتن نیرو هستید و می خواهید به استیت حمله بروید .
- **خروجی:**
- در صورت موفقیت آمیز بودن درخواست خروجی برابر با یک دیکشنری است که شامل ۲ کلید است:
- کلید game\_state که مقدار آن برابر استیت جدید بازی است که یکی از سه وضعیت : قرار دادن نیرو/حمله/انتقال نیرو است.
- کلید message که مقدار success دارد.

## fort()

- این تابع ۲ تا ورودی می‌گیرد:
- ورودی node\_id برابر با شماره سیاره است.
- ورودی troop\_count برابر با تعداد نیروهایی است که بازیکن می‌خواهد آن را به نیروی دفاعی تبدیل کند.
- در صورت موفقیت آمیز بودن عملیات خروجی یک دیکشنری است که کلید آن برابر با success است و مقدار آن برابر با there is not enough troops in the node است..



AICUP  
2023

[aicup\\_official](#)

[t.me/aicup](#)

[aicup2023.ir](#)

هوش مصنوعی را متفاوت تجربه کنید!