



بسمه تعالی

مستند مسابقه (مرحله حضوری)

فیردیه زبان جاوا ۹۲۱

بهمن و اسفند ۱۳۹۳



در قسمت قبل دیدیم که ...



از تیم سری مخوف، به تیم کد زن نوب

خبر تولید سلول های قابل کنترل دانشگاه شریف توی کشور پخش شده، این فرصت خوبیه برای شما که خودتون رو به ما اثبات کنید و ما هم قدرت اول دنیا رو به دست بیاریم. ما به سختی به محفظه ی سری محل نگهداری این سلول ها نفوذ کردیم و به شبکه ی کنترل این سلول ها دسترسی داریم و می خوایم که اون ها رو اونقدر زیاد و قوی بکنیم که بتونن از اون محفظه فرار کنن. حالا کاری که شما باید انجام بدید اینه که تا جای ممکن تعداد این سلول ها رو زیاد کنید و سطح انرژی اون ها رو بالا ببرید.

تیم کد زن نوب! جزئیات بیشتر همراه با پیوست براتون ارسال میشه.

و اینک ادامه داستان ...





آزمایشگاه A901 دانشگاه شریف | گزارش محرمانه شماره ۱۸



بررسی سلول‌های کنترل شونده، ویژگی‌های تازه‌ای از آن‌ها را مشخص کرد. در آزمایشات اخیر به این نتیجه رسیدیم که این سلول‌ها توانایی تقلید رفتار موجودات دیگر را دارند. در آزمایش اولیه، تعدادی از این سلول‌ها در مقابل تصویر یک ربات انسان‌نما قرار گرفتند و به سرعت شکل و رفتار آن را شبیه‌سازی کردند. با گذشت زمان قدرت این سلول‌ها در تطابق با محیط اطراف بهبود یافته است و طبق مشاهدات روند تکامل آن‌ها بسیار سریع است، به طوری که خطر از کنترل خارج شدن آن‌ها وجود خواهد داشت. بررسی رفتار این سلول‌ها ادامه دارد.

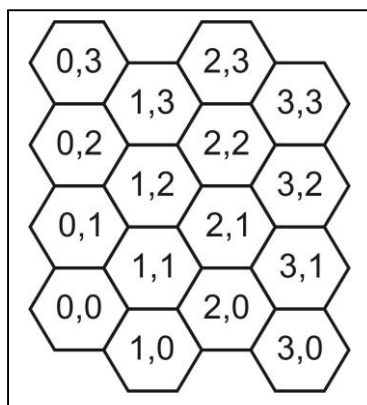
فهرست مطالب

۶	مقدمه
۶	نقشه
۶	بلوک‌ها
۷	سلول‌ها
۱۱	نکاتی در مورد بازی و داوری مرحله غیر حضوری
۱۲	نصب پیش نیازهای لازم
۱۳	راهنمای رابط برنامه‌نویسی
۲۲	آغاز
۲۳	نکات مهم
۲۴	ضمیمه ۱: NoobFriendly
۲۶	ضمیمه ۲: شروع به نوشتن کد در Java
۲۷	ضمیمه ۳: شروع به نوشتن کد در C++
۲۸	ضمیمه ۴: شروع به نوشتن کد در Python
۲۹	ضمیمه ۵: اضافه کردن JDK 1.8 به Eclipse

مقدمه

در این مرحله، بازی به صورت رقابتی انجام می‌شود و هر مسابقه بین ۲ تا ۴ تیم خواهد داشت. در هر مسابقه هر تیم باید تلاش کند که مجموع انرژی سلول‌های خود را افزایش دهد و رتبه بندی هر مسابقه بر اساس مجموع انرژی سلول‌های تیم‌ها در انتهای بازی است. بازی در نقشه‌ای که از بلوک (خانه)های ۶ ضلعی تشکیل شده است، انجام می‌شود. در ابتدای بازی، هر تیم چند سلول در اختیار دارد که در بلوک‌های نقشه قرار دارند. در ادامه این سلول‌ها با دریافت دستوراتی از کد تیم‌های شرکت‌کننده اعمالی را انجام می‌دهند. اجرای بازی به صورت نوبتی^۱ است و هر سلول در هر نوبت می‌تواند یکی از اعمال ممکن در بازی را انجام دهد که شامل حرکت به بلوک‌های مجاور، جمع‌آوری منبع، تقسیم سلول و حمله به سلول‌های دیگر می‌شود. البته دو عمل جمع‌آوری منبع و تقسیم سلول فقط در خانه‌های خاصی از نقشه می‌توانند انجام شوند. هر سلول فقط می‌تواند خانه‌هایی را ببیند که در دامنه‌ی دیدش هستند و شامل خانه‌هایی است که در نزدیکی محل سلول هستند. در انتهای بازی مجموع انرژی سلول‌های شما محاسبه شده و به عنوان امتیاز شما در نظر گرفته خواهد شد. سلول‌های این مرحله قابلیت جهش و تکامل در خانه‌های تقسیم را دارند. در ادامه جزئیات بازی به تفصیل توضیح داده می‌شود.

نقشه



مختصات نقشه بازی به صورت شکل زیر است که جنوب غربی ترین نقطه‌ی نقشه مختصات (0,0) دارد. مولفه اول نشان دهنده‌ی مولفه‌ی x و مولفه دوم نشان دهنده‌ی مولفه‌ی y است. در انتهای این مستند نقشه‌ای از این شش ضلعی‌ها وجود دارد که می‌توانید آن را چاپ کرده و با استفاده از آن فکر کنید.

بلوک‌ها

هر کدام از بلوک‌های نقشه ارتفاع خاصی دارند. ارتفاع بلوک‌ها در حرکت سلول تاثیر می‌گذارد و سلول نمی‌تواند به بلوکی خیلی بلندتر از بلوکی که در آن است، برود. بلوک‌ها به یکی از انواع زیر دیده می‌شوند:

- خانه نامشخص^۲: خانه‌ای که تا به حال در دامنه دید سلول‌های تیم قرار نگرفته‌اند و نوع آن‌ها مشخص نیست.

¹ Turn-based

² None

• خانه منبع^۳: خانه ای است که منبع در آن وجود دارد. اگر در نوبتی که یک سلول روی این خانه است دستور جمع آوری منبع به آن سلول داده شود مقدار مشخصی منبع را کسب می کند. مقدار مشخصی منبع در هر خانه ی منبع است و با جمع آوری منبع این مقدار در آن خانه کم شده و در نهایت تمام می شود.

به خانه های منبع، ارتفاعی علاوه بر ارتفاع اولیه شان، به عنوان ارتفاع منبع اضافه می شود. ارتفاع نهایی خانه های منبع به صورت زیر محاسبه می شود.

$$h = \min\left\{H + \frac{R}{50}, 9\right\}$$

که H ارتفاع اولیه خانه ی منبع، R مقدار منبع فعلی خانه و h ارتفاع فعلی خانه ی منبع است.

• خانه تقسیم^۴: خانه ای که عمل تقسیم (دو تا شدن) می تواند در آن انجام شود. اگر در نوبتی که یک سلول روی این خانه است دستور تقسیم به آن سلول داده شود در صورت وجود شرایط لازم سلول تقسیم شده و سلول جدید به خانه مجاور می رود. چهار ویژگی برای سلول ها وجود دارد که قابلیت زیاد شدن و جهش دارند. (این ویژگی ها در بخش سلول ها توضیح داده می شود) در هر خانه ی تقسیم صفر، یک یا چند تا از این ویژگی ها زیاد می شود و سلول تکامل پیدا می کند. توجه کنید که این تکامل و جهش فقط بر روی سلول جدید (فرزند) اعمال می شود و سلول قدیمی (والد) با همان ویژگی های قبلی باقی می ماند.

• خانه غیر قابل عبور^۵: خانه هایی هستند که رفتن به آن ها ناممکن است.

• خانه معمولی^۶: خانه های معمولی نقشه هستند که ویژگی های هیچ کدام از انواع فوق را ندارند.

سلول ها

هر تیم دارای تعدادی سلول است که در طول بازی سعی می کند مجموع انرژی آن ها را بیشینه کند. در این مستند همیشه منظور از سلول موجودات سلولی است و اشاره به خانه های نقشه ندارد. هر سلول دارای سطح انرژی است که میزان گلوکز ذخیره شده در سلول را نشان می دهد. این مقدار حداکثر برابر ۱۰۰ واحد خواهد بود. سلول ها قابلیت تکامل پیدا کردن در خانه های تقسیم را دارند. در ادامه ویژگی های قابل تکامل سلول ها را توضیح می دهیم.

• نرخ جذب^۷ سلول، سرعت جذب منابع از خانه منبع را مشخص می کند. اگر میزان منبع موجود در خانه ی منبع کمتر از ضریب جذب باشد، سلول کل منبع موجود را جذب می کند. کمینه مقدار نرخ جذب برابر ۱۵ است. حداکثر مقدار نرخ جذبی که یک سلول می تواند داشته باشد برابر ۴۵ است.

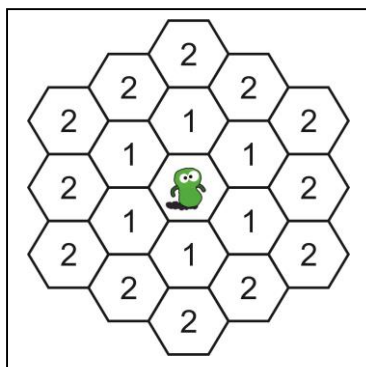
³ Resource

⁴ Mitosis

⁵ Impassable

⁶ Normal

⁷ Gain Rate



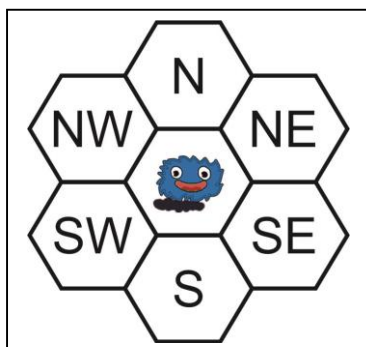
- شعاع دید هر سلول مشخص می کند که سلول تا چه فاصله ای از خانه های اطرافش را می بیند. کمینه مقدار شعاع دید برابر ۲ است به این معنی که می تواند خانه هایی را که با فاصله حداکثر ۲ از آن قرار دارند را ببیند. سلول می تواند نوع، ارتفاع و حضور یا عدم حضور یک سلول در بلوک هایی که می بیند را تشخیص دهد. در شکل روبرو می توانید خانه هایی که یک سلول می بیند (برای حالتی که شعاع دید برابر ۲ است) و فاصله ی آنها تا

سلول را مشاهده کنید. حداکثر مقدار شعاع دید یک سلول برابر ۵ است.

- ارتفاع پرش، مشخص می کند که سلول به خانه ی همسایه ای چقدر بلند تر از محل فعلی اش می تواند برود. کمینه مقدار ارتفاع پرش برابر ۲ است به این معنی که سلول می تواند از محل فعلی اش به خانه ای برود که حداکثر دو واحد بلندتر از خانه ی فعلی است. حداکثر ارتفاع پرش یک سلول برابر ۵ است.

- خسارت حمله، مقدار انرژی که با یک بار حمله از سلول حریف کم می شود را مشخص می کند. کمینه مقدار خسارت حمله برابر ۲۰ است به این معنی که با یک بار حمله به سلول حریف ۲۰ واحد از انرژی حریف کم می شود. حداکثر خسارت حمله یک سلول برابر ۳۵ است.

در هر نوبت شما می توانید به هر سلول یک دستور دهید و آن سلول در صورتی که شرایط لازم برای اجرای دستور وجود داشته باشد آن را انجام می دهد. در صورتی که چند دستور به یک سلول داده شود دستور آخر اجرا خواهد شد. دستورات ممکن در ادامه توضیح داده می شوند.



- حرکت^۸: دستور حرکت نیاز به یک جهت دارد که مشخص کننده ی یک خانه ی همسایه ی محل حضور سلول است. در مجموع شش جهت شمال^۹، شمال شرقی^{۱۰}، جنوب شرقی^{۱۱}، جنوب^{۱۲}، جنوب غربی^{۱۳} و شمال غربی^{۱۴} وجود دارد. در شکل روبرو خانه ی متناظر با هر کدام از این جهتها نشان داده شده است.

شرایط و نحوه ی انجام حرکات به صورت زیر است.

⁸ Move

⁹ North

¹⁰ North-East

¹¹ South-East

¹² South

¹³ South-West

¹⁴ North-West

- حرکت در صورتی انجام می‌شود که بلوک مقصد از نوع غیر قابل عبور نباشد، خالی باشد و ارتفاع آن حداکثر به اندازه ی ارتفاع پرش سلول بیشتر از خانه مبدا باشد و یا اینکه خانه ی مقصد هم ارتفاع یا کم ارتفاع تر از خانه ی مبدا باشد. (دقت کنید که حرکت فقط می‌تواند به بلوک‌های نقشه انجام شود و نمی‌توان از نقشه خارج شد!)
- در صورتی که دو سلول به صورت هم زمان دستور حرکت به یک خانه را صادر کنند، در نوبت بعدی، یکی از این سلول‌ها (به صورت تصادفی) در خانه‌ی مقصد قرار گرفته و سلول دیگر حرکتی انجام نمی‌دهد.
- اگر سلولی بخواهد به خانه ای حرکت کند که در این نوبت خالی می‌شود (یعنی سلولی که در خانه ی مقصد است بخواهد آن خانه را ترک کند)، حرکت انجام خواهد شد. (برخلاف مرحله غیر حضوری) حتی در حالتی که دو سلول در دو خانه ی مجاور بخواهند با یکدیگر جابجا شوند این اتفاق می‌افتد.
- تقسیم^{۱۵}: با ارسال این دستور در صورت وجود شرایط زیر سلولی که دستور را دریافت کرده به دو سلول تبدیل می‌شود و سلول فرزند ویژگی‌هایی را که خانه ی تقسیم اضافه می‌کند را دریافت می‌کند.
 - عمل تقسیم سلولی تنها در بلوک‌های تقسیم قابل انجام است و در بلوک‌های دیگر ارسال در صورت ارسال این دستور اتفاقی نمی‌افتد.
 - عمل تقسیم، تنها در صورتی انجام می‌شود که در هنگام صدور دستور، سلول جدید بتواند به یکی از خانه‌های همسایه برود که شرایط آن مانند شرایط حرکت سلول است. سلول جدید به صورت تصادفی به یکی از خانه‌هایی که شرایط حرکت به آن‌ها وجود دارد، می‌رود.
 - سطح انرژی سلول در هنگام صدور دستور تقسیم باید بالاتر از ۸۰ باشد و اگر کمتر باشد اتفاقی نخواهد افتاد.
 - پس از انجام عمل تقسیم، دو سلول با سطح انرژی ۴۰ خواهیم داشت.
 - توجه کنید که تقسیم بر حرکت ارجحیت دارد. اگر به یک سلول دستور تقسیم داده شود و خانه ای که برای فرزند آن انتخاب می‌شود، خانه ای باشد که سلول دیگری در این نوبت می‌خواهد با ارسال دستور حرکت به آن وارد شود، دستور حرکت انجام نمی‌شود و عمل تقسیم انجام خواهد شد.
- جمع آوری منبع^{۱۶}: در صورتی که سلول در یک خانه ی منبع باشد و این دستور داده شود، مقداری انرژی از خانه ی منبع کم شده و به سلول اضافه می‌شود. در صورتی که مقدار انرژی خانه منبع بیشتر یا مساوی

¹⁵ Mitosis

¹⁶ Gain Resource

نرخ جذب باشد مقدار انرژی که جمع آوری می‌شود برابر نرخ جذب است و در غیر این صورت به اندازه ی انرژی موجود در خانه ی منبع است.

- حمله^{۱۷}: دستور حمله نیز مانند حرکت نیاز به یک جهت دارد. در صورتی که به سمت خانه ای که در آن یک سلول وجود دارد حمله شود، به اندازه ی میزان خسارت حمله ی حمله کننده از انرژی سلولی که مورد حمله قرار گرفته است کم می‌شود و اگر انرژی اش از خسارت حمله کننده کمتر باشد آن سلول می‌میرد. توجه کنید که Friendly Fire روشن (on) است!

اولویت اعمال سلول ها از زیاد به کم اینگونه است: تقسیم، حمله، جمع آوری منبع و حرکت.

نکاتی در مورد بازی و داوری مرحله حضوری

در این قسمت توضیحاتی در مورد بازی مرحله ی حضوری و رقابت های آن داده می شود.

نکاتی در مورد بازی

- هر بازی ۵۰۰ نوبت طول می کشد.
- حداکثر اندازه نقشه ۶۰×۴۰ است.
- امتیاز شما در یک بازی به صورت مجموع انرژی سلول هایتان محاسبه می شود.
- نحوه ی اجرای بازی ها و صعود در رقابت ها متعاقبا اعلام می شود.
- برای اجرای بازی ها حتما باید دو client اجرا شوند. برای client دوم یک فایل jar به شما داده شده است. توجه کنید که این client صرفا الگوریتم تصادفی پیش فرض است.
- نقشه هایی که در مرحله ی غیر حضوری استفاده می شده اند، در این مرحله قابل استفاده نیستند و باید از نقشه های جدید استفاده کنید.
- کلید های بالا، پایین، چپ و راست صفحه کلید برای چرخش و بزرگنمایی گرافیک استفاده می شوند. دکمه های W ، A ، S و D برای حرکت دوربین استفاده می شوند و دکمه ی R برای بازگشت به حالت پیش فرض دوربین است.
- نقشه هایی که کدها روی آن ها اجرا می شود قطعا با نقشه های نمونه ای که به شما داده شده است متفاوت است.
- بیشتر بازی های مرحله ی حضوری به صورت چهار تیمه برگزار خواهد شد.
- قانون خاصی برای محل و نوع خانه های مختلف نقشه وجود ندارد و ممکن است که در نقشه هر نوع خانه ای در هر محلی باشد.
- نقشه های مورد استفاده به جهت داوری برای تمام تیم ها یکسان هستند.

نکاتی در مورد محدودیت های اجرا

- قدرت پردازشی که در اختیار کد شما قرار می گیرد 1GH است.
- میزان حافظه ای که در اختیار کد شما قرار می گیرد 500MB است.

نصب پیش نیازهای لازم

برای اجرای بازی و نوشتن کدهای خود نیاز به نصب پیش نیازهای زیر دارید.

1. Java Development Kit 1.8.0
2. Python 3
3. C++ 11
4. C++ Boost 1.55

مورد ۱ برای اجرای بازی ضروری است. مورد ۲ فقط در صورتی لازم است بخواهید با Python کد بنویسید و موارد ۳ و ۴ فقط در صورتی لازم هستند که بخواهید با C++ کد بنویسید. توجه کنید که از زبان C++ فقط تحت سیستم عامل لینوکس پشتیبانی می‌شود. در ادامه نحوه نصب هر کدام از این پیش نیازها توضیح داده خواهد شد.

Java Development Kit 1.8.0 (۱)

برای دریافت این پیش نیاز می‌توانید به لینک زیر مراجعه کنید و فایل سازگار با سیستم عامل خود را دریافت

کنید.

<http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>

برای دیدن راهنمای نصب این پیش نیاز در سیستم عامل‌های مختلف می‌توانید به لینک‌های زیر مراجعه کنید.

Windows	http://www.wikihow.com/Install-the-Java-Software-Development-Kit
Linux	http://www.wikihow.com/Install-Oracle-Java-JDK-on-Ubuntu-Linux
Mac-OS	http://www.wikihow.com/Install-the-JDK-(Java-Development-Kit)-on-Mac-OS-X

Python 3 (۲)

نصب این پیش نیاز برای گروه‌هایی لازم است که قصد استفاده از Client پایتون را دارند.

برای دریافت این پیش نیاز، با توجه به سیستم عامل خود به یکی از لینک‌های زیر مراجعه کنید.

Windows	https://www.python.org/downloads/windows/
Linux	https://www.python.org/downloads/source/
Mac-OS	https://www.python.org/downloads/mac-osx/

برای دیدن راهنمای نصب این پیش نیاز می‌توانید به لینک زیر مراجعه کنید.

www.wikihow.com/Install-Python

C++11 Compiler (۳)

گروه‌هایی که قصد استفاده از Client زبان C++ را دارند، باید یک کامپایلر با قابلیت کامپایل کردن کدهای

C++11 را داشته باشند.

Client این زبان روی سیستم عامل لینوکس کامپایل می‌شود.

C++ Boost 1.55 (۴)

گروه‌هایی که قصد کار با زبان C++ را دارند لازم است که کتابخانه‌ی Boost نسخه‌ی ۱,۵۵ را نیز روی سیستم

خود داشته باشند. این کتابخانه را می‌توانید از لینک زیر دانلود کنید:

http://www.boost.org/users/history/version_1_55_0.html

راهنمای رابط برنامه‌نویسی

کلاسی به نام AI در اختیار شما قرار داده شده است. در این کلاس تابعی به نام doTurn وجود دارد. در هر نوبت، این تابع یک بار صدا زده می‌شود و شما باید در این تابع دستوراتی را که می‌خواهید به سلول‌های خود بدهید. پس شما باید این تابع را به گونه‌ای بنویسید که سلول‌هایتان هوشمندانه عمل کنند. هر نوبت بازی یک ثانیه طول می‌کشد که شما در ۴۰۰ میلی ثانیه اول آن فرصت دارید که دستورات لازم را به سلول‌هایتان بدهید. تابع doTurn به عنوان ورودی یک شیء از نوع کلاس World می‌گیرد. این شیء بیانگر حالت کنونی جهان است و درون آن، تمامی اطلاعاتی قرار دارد که شما به آن‌ها دسترسی دارید. در زیر شکل کلی این کلاس در زبان‌های متفاوت را مشاهده می‌کنید. کسانی که از Python استفاده می‌کنند توجه کنند که تمام مواردی که لازم دارند در فایل Model.py است و اطلاعات آن کافی است. برای اطلاعات بیشتر می‌توانید به مطالب زیر مراجعه کنید.

Java	C++	Python
<pre>class AI { doTurn(World world) { // AI code comes here } }</pre>	<pre>class AI { void doTurn(World* world) { // AI code comes here } };</pre>	<pre>class AI(): def doTurn(world): // AI code comes here</pre>

کلاس World

این کلاس شامل اطلاعاتی از شرایط حال حاضر بازی است. دقت کنید که در هر نوبت این اطلاعات تغییر می‌کنند و اگر شما می‌خواهید که به اطلاعات گذشته دسترسی داشته باشید باید خودتان آن را ذخیره کنید. نحوه دسترسی به اطلاعاتی از این کلاس که می‌توانید استفاده کنید را در ادامه توضیح می‌دهیم.

:Teams

دسترسی به اسامی تیم‌های درون بازی به صورت آرایه‌ای از رشته‌ها.

Java	C++	Python
String[] getTeams()	std::vector<std::string> getTeams()	World.teams

:My Id

دسترسی به شماره تیم خود.

Java	C++	Python
int getMyId()	int getMyId()	World.my_id

:My Name

دسترسی به نام تیم خود.

Java	C++	Python
String getName ()	std::string getName ()	World.my_name

:Map Size

دسترسی به اندازه ی نقشه در قالب یک شیء از نوع MapSize.

Java	C++	Python
MapSize getSize ()	MapSize getSize ()	World.map_size

:Map

دسترسی به نقشه ی کنونی بازی و اطلاعات آن.

Java	C++	Python
Map getMap ()	Map* getMap ()	World.map

:Turn

دسترسی به شماره نوبت کنونی بازی.

Java	C++	Python
int getTurn ()	int getTurn ()	World.turn

:All Cells

دسترسی به تمام سلول های موجود در دامنه دید سلول های تیم خود در قالب یک لیست از Cell ها

Java	C++	Python
ArrayList<Cell> getAllCells ()	std::vector<Cell*> getAllCells ()	World.all_cells

:My Cells

دسترسی به سلول های تیم در قالب یک لیست از Cell ها

Java	C++	Python
ArrayList<Cell> getMyCells ()	std::vector<Cell*> getMyCells ()	World.my_cells

:Enemy Cells

دسترسی به سلول‌های تیم در قالب یک لیست از Cellها

Java	C++	Python
ArrayList<Cell> getEnemyCells()	std::vector<Cell*> getEnemyCells()	World.enemy_cells

MapSize

شیء MapSize شامل دو بخش برای طول و عرض نقشه است. که به صورت زیر قابل دسترسی هستند.

Java	C++	Python
int getHeight() int getWidth()	int getHeight() int getWidth()	MapSize["height"] MapSize["width"]

کلاس Map

در این کلاس اطلاعات و توابع مربوط به بلوک‌های نقشه وجود دارد.

:at

این تابع با گرفتن ورودی به صورت Position یا x و y بلوکی با آن مختصات را بر می‌گرداند.

Java	C++	Python
Block at(Position) Block at(int x, int y)	Block* at(Position) Block* at(int x, int y)	at(pos)

کلاس Cell

هر شیء از این کلاس یکی از سلول‌های موجود است که اطلاعات و توابع زیر را دارد.

:id

دسترسی به شماره سلول.

Java	C++	Python
int getId()	int getId()	Cell.id

:Position

دسترسی به محل سلول در قالب یک شیء از کلاس Position.

Java	C++	Python
Position getPos()	Position getPos()	Cell.pos

:Team Id

دسترسی به شماره تیم مربوط به سلول

Java	C++	Python
<code>int getTeamId()</code>	<code>int getTeamId()</code>	<code>Cell.team_id</code>

:Energy

دسترسی به انرژی فعلی سلول

Java	C++	Python
<code>int getEnergy()</code>	<code>int getEnergy()</code>	<code>Cell.energy</code>

:Depth of Field

دسترسی به اندازه میدان دید سلول (اگر این مقدار را بخواهید برای سلول های حریف ببینید، مقداری نامرتبط با

مقدار اصلی بر می گرداند.)

Java	C++	Python
<code>int getDepthOfField()</code>	<code>int getDepthOfField()</code>	<code>Cell.depth_of_field</code>

:Jump Height

دسترسی به ارتفاع پرش سلول (اگر این مقدار را بخواهید برای سلول های حریف ببینید، مقداری نامرتبط با مقدار

اصلی بر می گرداند.)

Java	C++	Python
<code>int getJump()</code>	<code>int getJump()</code>	<code>Cell.jump</code>

:Gain Rate

دسترسی به نرخ جذب سلول (اگر این مقدار را بخواهید برای سلول های حریف ببینید، مقداری نامرتبط با مقدار

اصلی بر می گرداند.)

Java	C++	Python
<code>int getGainRate()</code>	<code>int getGainRate()</code>	<code>Cell.gain_rate</code>

:Damage

دسترسی به خسارت حمله سلول (اگر این مقدار را بخواهید برای سلول های حریف ببینید، مقداری نامرتبط با

مقدار اصلی بر می گرداند.)

Java	C++	Python
<code>int getAttack()</code>	<code>int getAttackValue()</code>	<code>Cell.attack_value</code>

:move

برای حرکت دادن سلول می‌توانید با دادن یکی از شش جهت موجود در کلاس Direction دستور حرکت به آن جهت را با استفاده از این تابع به سلول بدهید.

Java	C++	Python
void move(Direction)	void move(Direction)	move(direction)

:gainResource

با استفاده از این تابع می‌توانید دستور جذب منبع را به سلول بدهید.

Java	C++	Python
void gainResource()	void gainResource()	gain_resource()

:mitosis

با استفاده از این تابع می‌توانید دستور تقسیم را به سلول بدهید.

Java	C++	Python
void mitosis()	void mitosis()	mitosis()

:attack

برای حمله کردن سلول می‌توانید با دادن یکی از شش جهت موجود در کلاس Direction دستور حمله به آن جهت را با استفاده از این تابع به سلول بدهید.

Java	C++	Python
void attack(Direction)	void attack(Direction)	attack(direction)

Direction

یک Enum (در C++ و Java) و یا کلاس (در Python) است که شش جهت ممکن برای حرکت در آن قرار دارد

که به صورت زیر قابل دسترسی هستند.

Java	C++	Python
Direction.NORTH	Direction::NORTH	Constants.Direction.NORTH
Direction.NORTH_EAST	Direction::NORTH_EAST	Constants.Direction.NORTH_EAST
Direction.SOUTH_EAST	Direction::SOUTH_EAST	Constants.Direction.SOUTH_EAST
Direction.SOUTH	Direction::SOUTH	Constants.Direction.SOUTH
Direction.SOUTH_WEST	Direction::SOUTH_WEST	Constants.Direction.SOUTH_WEST
Direction.NORTH_WEST	Direction::NORTH_WEST	Constants.Direction.NORTH_WEST

Position

مختصات موقعیت مربوطه و موقعیت‌های جدیدی که از یک خانه با یک جهت می‌توان به آن‌ها رفت از این طریق قابل دستیابی هستند.

:Next Position

دسترسی به موقعیت^{۱۸} خانه ی همسایه یک موقعیت که در جهت ورودی داده شده است.

Java	C++	Python ^{۱۹}
Position getNextPos (Direction)	Position getNextPos (Direction)	get_next_pos (direction , position)

:X

دسترسی به مختصات X یک موقعیت.

Java	C++	Python
int getX ()	int getX ()	position["x"]

:Y

دسترسی به مختصات Y یک موقعیت.

Java	C++	Python
int getY ()	int getY ()	position["y"]

کلاس Block

این کلاس تمام اطلاعات مربوط به یک بلوک را در خود نگه می‌دارد.

:Position

برای پیدا کردن موقعیت بلوک می‌توانید از این تابع استفاده کنید.

Java	C++	Python
Position getPos ()	Position getPos ()	Block.pos

:Minimum Height

برای پیدا کردن ارتفاع اولیه بلوک می‌توانید از این تابع استفاده کنید. البته توجه کنید که مقدار ارتفاع و ارتفاع اولیه فقط برای خانه‌های منبع متفاوت است.

Java	C++	Python
int getMinHeight ()	int getMinHeight ()	Block.min_height

¹⁸ Position

^{۱۹} این تابع برای Python در کلاس Map است.

:Height

برای پیدا کردن ارتفاع بلوک می‌توانید از این تابع استفاده کنید. البته توجه کنید که مقدار ارتفاع و ارتفاع اولیه فقط برای خانه های منبع متفاوت است.

Java	C++	Python
<code>int getHeight()</code>	<code>int getHeight()</code>	<code>Block.height()</code>

:Resource

برای پیدا کردن میزان منبع یک بلوک می‌توانید از این تابع استفاده کنید. توجه کنید که این تابع برای خانه‌های غیر منبع نیز وجود دارد که برای آن‌ها مقدار صفر را بر می‌گرداند.

Java	C++	Python
<code>int getResource()</code>	<code>int getResource()</code>	<code>Block.resource</code>

:Type

برای پیدا کردن نوع بلوک می‌توانید از این تابع استفاده کنید.

Java	C++	Python
<code>String getType()</code>	<code>std::string getType()</code>	<code>Block.type</code>

:Gain Rate Improvement Amount

دسترسی به میزان افزایش نرخ جذب خانه ی تقسیم. (توجه کنید که این مقدار فقط برای خانه های تقسیم است.)

Java	C++	Python
<code>int getGainImprovementAmount()</code>	<code>int getGainImprovementAmount()</code>	<code>Block.gain_improvement_amount</code>

:Depth of Field Improvement Amount

دسترسی به میزان افزایش دامنه ی دید خانه ی تقسیم. (توجه کنید که این مقدار فقط برای خانه های تقسیم است.)

Java	C++	Python
<code>int getDepthOfFieldImprovementAmount()</code>	<code>int getDepthOfFieldImprovementAmount()</code>	<code>Block.depth_of_field_improvement_amount</code>

:Jump Improvement Amount

دسترسی به میزان افزایش ارتفاع پرش خانه ی تقسیم. (توجه کنید که این مقدار فقط برای خانه های تقسیم است.)

Java	C++	Python
<code>int getJumpImprovementAmount()</code>	<code>int getJumpImprovementAmount()</code>	<code>Block.jump_improvement_amount</code>

:Damage Improvement Amount

دسترسی به میزان افزایش خسارت حمله ی خانه ی تقسیم. (توجه کنید که این مقدار فقط برای خانه های تقسیم است.)

Java	C++	Python
int getAttackImprovement Amount ()	int getAttackImprovement Amount ()	Block.attack_improvement_amount

کلاس Constants

در این کلاس مقادیر ثابت بازی و رشته‌هایی ثابت با محتوای نوع خانه‌ها ذخیره شده است.

:None Block

رشته ای با محتوای "none".

Java	C++	Python
String BLOCK_TYPE_NONE	std::string BLOCK_TYPE_NONE	Constants.BLOCK_TYPE_NONE

:Normal Block

رشته ای با محتوای "normal".

Java	C++	Python
String BLOCK_TYPE_NORMAL	std::string BLOCK_TYPE_NORMAL	Constants.BLOCK_TYPE_NORMAL

:Mitosis Block

رشته ای با محتوای "mitosis".

Java	C++	Python
String BLOCK_TYPE_MITOSIS	std::string BLOCK_TYPE_MITOSIS	Constants.BLOCK_TYPE_MITOSIS

:Resource Block

رشته ای با محتوای "resource".

Java	C++	Python
String BLOCK_TYPE_RESOURCE	std::string BLOCK_TYPE_RESOURCE	Constants.BLOCK_TYPE_RESOURCE

:Impassable Block

رشته ای با محتوای "impassable".

Java	C++	Python
String BLOCK_TYPE_IMPASSABLE	std::string BLOCK_TYPE_IMPASSABLE	Constants.BLOCK_TYPE_IMPASSABLE

:Min Energy for Mitosis

مقدار حداقل انرژی مورد نیاز برای انجام عمل تقسیم که برابر ۸۰ است.

Java	C++	Python
int CELL_MIN_ENERGY_FOR_MITOSIS	int CELL_MIN_ENERGY_FOR_MITOSIS	Constants.CELL_MIN_ENERGY_FOR_MITOSIS

:Max Energy

بیشینه مقدار انرژی که یک سلول می‌تواند داشته باشد که برابر ۱۰۰ است.

Java	C++	Python
int CELL_MAX_ENERGY	int CELL_MAX_ENERGY	Constants.CELL_MAX_ENERGY

:Gain Rate

مقدار کمینه منبعی که یک سلول می‌تواند در یک نوبت دریافت کند که برابر ۱۵ است.

Java	C++	Python
int CELL_GAIN_RATE	int CELL_GAIN_RATE	Constants.CELL_GAIN_RATE

:Depth of Field

کمینه اندازه ی میدان دید سلول که برابر ۲ است.

Java	C++	Python
int CELL_DEPTH_OF_FIELD	int CELL_DEPTH_OF_FIELD	Constants.CELL_DEPTH_OF_FIELD

آغاز

برای شروع به کد زنی و اجرای برنامه خود شما باید برنامه JavaChallenge2015 را اجرا کنید که در داخل فایل فشرده ی NoobFriendly به شما داده خواهد شد. این برنامه در واقع سرور برنامه را اجرا می کند. توضیحات لازم برای اجرای این برنامه در ضمیمه ۱ مستند داده شده است. بقیه ی ضمایم نیز در مورد شروع کد زنی برای برنامه ی client است که با توجه به زبان برنامه نویسی که انتخاب می کنید و مشکلاتی که به آن ها بر می خورید می توانید به آن ها مراجعه کنید.

نکات مهم

- شما فقط می توانید برای یکی از زبان های C++، Java و Python کد ارسال کنید و امکان ارسال کد برای چند زبان وجود ندارد.
- در صورت داشتن سوال یا بروز هر گونه مشکل به سامانه پرسش و پاسخ در آدرس <http://javachallenge.ir/qa/> مراجعه کنید و سوال خود را در آن صفحه مطرح کنید.
- برای پرسیدن سوال در سامانه پرسش و پاسخ، حتما از برچسب هایی در مورد زبان برنامه نویسی مورد استفاده (C++، Java و python) و سیستم عامل (Windows، Linux و Mac OS X) خود استفاده کنید.
- لطفا قبل از ارسال سوال در سامانه ی پرسش و پاسخ از غیر تکراری بودن آن اطمینان حاصل کنید.
- کلاس ها و توابعی که در طول مستند بررسی شدند بیشتر مورد نیاز هستند، ولی برای آشنایی بیشتر با نحوه عملکرد می توانید به خود کد مراجعه کنید.
- نحوه ی ارسال پاسخ ها متعاقبا اعلام خواهد شد.
- برای دسترسی به اینترنت، یک اتصال LAN به هر تیم داده شده است و تیم ها می توانند بوسیله ی شبکه ی خصوصی مجازی^{۲۰} شناسه ی شریف^{۲۱} به اینترنت متصل شوند. برای انجام این کار به راهنمای موجود در صفحه ی <http://id.sharif.ir> مراجعه کنید و از نام کاربری و گذرواژه ای که در اختیارتان قرار داده شده است استفاده کنید.

²⁰ VPN²¹ Sharif ID

ضمیمه ۱: NoobFriendly

بازی ای که در اختیار شما قرار می‌گیرد دارای دو قسمت اصلی Server و Client است. بازی روی سرور اجرا می‌شود. در فایل‌هایی که برای شما آماده شده است، فایل Server نیز قرار دارد تا بتوانید کدهای خود را قبل از ارسال، تست کنید.

به منظور اجرای بازی، شما باید کارهای زیر را انجام دهید.

فایل NoobFriendly را Extract کنید، داخل آن با توجه به سیستم عامل خود یک فایل با نام JavaChallenge2015.bat (برای Windows) یا JavaChallenge2015.sh (برای Linux و Mac OS X) مشاهده می‌کنید. این فایل را اجرا کنید. اگر در سیستم عامل لینوکس با خطای Permission denied رو برو شدید دستور زیر را اجرا کنید.

```
chmod -R 777 *
```

اگر در سیستم عامل‌های زیر با اجرای فایل JavaChallenge2015 خطای پیدا نشدن فایل libudev.so.0 دریافت کردید به فولدر NOOBFRIENDLY_PATH\dest\Terminal_UI\node-webkit-v0.11.6-linux-x64 رفته و فایل bug-fix.sh را اجرا کنید. (انجام این کار تنها برای اولین بار کافی است).

- Ubuntu 13.04+

- Fedora 18+

- Arch

- Gentoo

- و مشتقات هر کدام از موارد بالا

اگر در هنگام اجرای JavaChallenge2015 با خطای Connection Failed مواجه شدید و با زدن چند باره ی دکمه Try Again مشکل رفع نشد، احتمالاً یکی از اشکالات زیر وجود دارد.

- JDK 1.8 به درستی نصب نشده یا به درستی به path اضافه نشده است. برای نصب درست JDK 1.8 به آموزش داده شده در صفحه ی ۷ مستند مراجعه کنید.

- یکی از port های مورد استفاده ی server باز است. ساده ترین راه برای بررسی وجود این مشکل restart کردن کامپیوتر است. اگر با restart کردن مشکل رفع شد از این پس بعد از اجرای هر بازی از بسته شدن کامل server و client اطمینان حاصل کنید.

اگر با رفع مشکلات بالا، همچنان با خطای Connection Failed مواجه شدید مشکل خود را در سامانه پرسش و پاسخ مطرح کنید.

پس از اجرای JavaChallenge2015 یک محیط گرافیکی مشاهده می‌کنید، در این قسمت روی آیکن Choose Map کلیک کرده و Map مورد نظر خود را انتخاب کنید (تعدادی Map در فولدر NOOBFRIENDLY_PATH\dest\map وجود دارد که می‌توانید از آن‌ها به منظور تست کردن کدهای خود استفاده کنید).

حال روی New Game کلیک کنید.

در این مرحله، Server منتظر می ماند تا شما کد خود را Run کنید. نحوه ی شروع به نوشتن کد و اجرای client در زبان های برنامه نویسی مختلف در ضمیمه های بعدی توضیح داده می شود. (کد خود را به صورت مجزا Run کنید و نگران ارتباط آن با Server نباشید، این قسمت از کار را ما به عهده می گیریم.)
پس از Run کردن کد خود، روی دکمه ی Start Game کلیک کنید و از بازی لذت ببرید.

تغییرات زیر را باید در NoobFriendly اعمال کنید.

Mitosis.jar در آدرس

NoobFriendly_[OS_NAME]/dest/

جایگزین شود.

فایل JGTerminal.nw در آدرس

NoobFriendly_[OS_NAME]/dest/Terminal_UI/

جایگزین شود.

فولدر resources در آدرس

NoobFriendly_[OS_NAME]/dest/

جایگزین شود.

ضمیمه ۲: شروع به نوشتن کد در Java

طریقه کد نویسی

- شما باید کد هوش مصنوعی خود را در تابع doTurn در فایل AI.java قرار دهید.
- شما مجاز هستید که فایل‌های موجود در پکیج client به جز Main.java را تغییر دهید یا فایل‌هایی را به این پکیج اضافه کنید.

نحوه اجرا

۱. در صورتی که JDK 1.8 را نصب کرده اید ولی در eclipse خود به آن دسترسی ندارید به ضمیمه ۵ مراجعه کنید.
۲. فولدر MitosisClient را داخل IDE خود، به عنوان مثال Eclipse، Import کنید. برای این کار به منوی File رجوع کرده و گزینه Import را انتخاب کنید. سپس در پنجره باز شده در پوشه General گزینه ی Existing Projects into Workspaces را انتخاب کنید و سپس Next را کلیک کنید. در صفحه ی جدید گزینه ی Select archive file را انتخاب کنید و آدرس فایل MitosisClient.zip را وارد کنید. سپس بر روی Finish کلیک کنید. حال باید libraryهای استفاده شده را به پروژه اضافه کنیم. برای این کار در قسمت Package Explorer (سمت چپ eclipse) در فولدر libs روی فایل gson-2.3.1.jar راست کلیک کرده و در زیرمنوی Build Path، گزینه ی Add to Build Path را انتخاب نمایید.
۳. پس از Import کردن این فولدر، یک بار آن را Run کنید تا از Build شدن آن اطمینان حاصل کنید. در صورتی که درست Build شود، ممکن است Exception ای با محتوای Connection Refused یا Error While Connecting to Server مشاهده کنید که به علت در حال اجرا نبودن سرور است.

ارسال فایل ها

برای ارسال فایل های خود، پکیج client را zip کنید و نام فایل فشرده را client.zip بگذارید. فایلی که شما ارسال می کنید فایل client.zip است.

ضمیمه ۳: شروع به نوشتن کد در C++

طریقه کد نویسی

- شما باید کد هوش مصنوعی خود را در تابع doTurn در فایل AI.cpp قرار دهید.
- شما مجاز هستید که تمام فایل هایی که برای کلاینت C++ به شما داده شده است به غیر از main.cpp را تغییر دهید یا فایل هایی را به آن اضافه کنید.

نحوه اجرا

برای نصب client مربوط به C++ در Linux ابتدا باید package های زیر را با استفاده از وارد کردن آن ها در terminal نصب کنید.

Compiler C++11:	sudo apt-get install build-essentials
Eclipse C++:	sudo apt-get install eclipse-cdt
Boost Library :	sudo apt-get install libboost-all-dev

برای اجرای پروژه C++ Client باید فایل zip ضمیمه شده به نام Cpp_Client.zip را در workspace برنامه unzip کنید و سپس در eclipse در قسمت Import Project در قسمت General گزینه Existing projects into Workspace را انتخاب کنید.

سپس در قسمت Select Root Directory فولدري که در workspace آنزيب کرديد را انتخاب کنید فولدر (JC) و گزینه Finish را بزنید.

حال می توانید project ایجاد شده را compile و اجرا کنید. ممکن است Exception ای با محتوای Connection Refused یا Error While Connecting to Server مشاهده کنید که به علت در حال اجرا نبودن سرور است.

کاربرانی که از زبان C++ استفاده می کنند، برای فرستادن کد خود در جاج مسابقات تنها می توانند از یکی از ۲ راه زیر استفاده کنند:

(۱) درست کردن فایل Makefile برای پروژه و فرستادن آن به همراه کدها. نام فایل خروجی که Makefile شما تولید می کند باید out باشد.

(۲) export کردن پروژه توسط eclipse به صورت General->Archive File. در این حالت کدها، فایل با پسوند project و فایل با پسوند cproject را برای export کردن باید انتخاب کنید و نیازی به فرستادن بقیه موارد نیست.

ارسال فایل ها

برای ارسال فایل های خود، تمامی فایل های مربوط به client را به همراه Makefile خود zip کنید و نام فایل فشرده را client.zip بگذارید. فایلی که شما ارسال می کنید فایل client.zip است.

ضمیمه ۴: شروع به نوشتن کد در Python

طریقه کد نویسی

- شما باید کد هوش مصنوعی خود را در تابع `do_turn` در فایل `AI.py` قرار دهید.
- شما مجاز هستید که تمام فایل هایی که برای کلاینت Python به شما داده شده است به غیر از `Controller.py` را تغییر دهید یا فایل هایی را به آن اضافه کنید.

نحوه اجرا

در ابتدا با استفاده از دستور العمل گفته شده در ضمیمه ۱، سرور را اجرا کنید و سپس با توجه به نوع سیستم عامل خود به یکی از روش های زیر `client` خود را اجرا کنید.

- **Windows:** فایل `Contoller.py` را اجرا کنید. در صورتی که با اجرای `Controller.py` مشکل دارید و Python 2 را از قبل نصب داشته اید، می توانید با کلیک راست روی فایل `Controller.py` به قسمت `open with` بروید و با انتخاب `python.exe` از مسیر نصب Python 3 کد را اجرا کنید. (مسیر پیش فرض `C:\Python34` است).
- **Linux:** ترمینال را باز کرده و وارد پوشه ی `python-client` بشوید. سپس با اجرای دستور زیر کد را اجرا کنید.

`Python3 Controller.py`

منتظر بمانید تا پیام `connected to server` چاپ شود. اگر پیام خطا چاپ شد، اطمینان حاصل کنید که سرور در حال اجرا است.

حال دکمه ی `start` را در `JavaChallenge2015` بزنید.

ارسال فایل ها

برای ارسال فایل های خود، تمامی فایل های مربوط به `client` را `zip` کنید و نام فایل فشرده را `client.zip` بگذارید. فایلی که شما ارسال می کنید فایل `client.zip` است.

ضمیمه ۵: اضافه کردن JDK 1.8 به Eclipse

توجه کنید که اگر از eclipse نسخه^{۲۲} Luna استفاده می‌کنید به این راهنما احتیاجی ندارید، ولی اگر از نسخه Kepler استفاده می‌کنید به راهنمایی زیر مراجعه کنید. نسخه‌های دیگر eclipse از JDK 1.8 پشتیبانی نمی‌کنند و اگر نسخه قدیمی تری دارید باید یکی از این دو نسخه را دانلود کنید.

ابتدا از نوار بالایی گزینه ی help را انتخاب کرده و سپس زیر منوی Eclipse Marketplace را انتخاب کنید. در فیلد Find عبارت java 8 را وارد کنید و java 8 support for Eclipse.. را نصب کنید. سپس eclipse خود را restart کنید.

بر روی پروژه کلیک راست کنید و گزینه ی properties را انتخاب کنید. سپس از منوی سمت چپ java compiler را انتخاب کنید و compliance level آن را به ۱٫۸ تغییر دهید.

حال بر روی java build path از منوی سمت چپ properties کلیک کنید و در صورتی که jre 1.8 دچار مشکل بود روی آن کلیک کنید و دکمه ی Edit را بزنید و دکمه ی installed JREs را انتخاب کنید. در صورتی که jre 1.8 وجود نداشت دکمه ی add را بزنید و پس از انتخاب Standard VM دکمه ی next را زده و پس از آن آدرس jre 1.8 خود را با زدن دکمه ی Directory وارد کنید. سپس بر روی Finish کلیک کنید.

