



بسمه تعالی

مستند مسابقه

فیردیه زبان جاوا ۹۲۱

بهمن و اسفند ۱۳۹۳



پردیس اول
PARDIS
AVAL



بازار 
cafebazaar.ir



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.

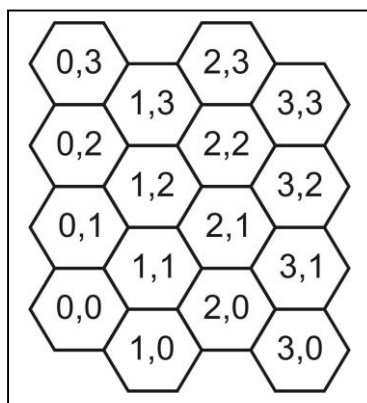
فهرست مطالب

۳	مقدمه
۳	نقشه
۳	بلوک‌ها
۴	سلول‌ها
۷	نصب پیش نیازهای لازم
۸	راهنمای رابط برنامه نویسی
۱۵	آغاز
۱۶	نکات مهم
۱۷	ضمیمه ۱: نحوه ی کار با NoobFriendly
۱۸	ضمیمه ۲: شروع به نوشتن کد در Java
۱۹	ضمیمه ۳: شروع به نوشتن کد در C++
۲۰	ضمیمه ۴: شروع به نوشتن کد در Python
۲۱	ضمیمه ۵: اضافه کردن JDK 1.8 به Eclipse

مقدمه

در این مرحله، بازی به صورت یک تیمه انجام می‌شود و هدف پیشینه کردن مجموع انرژی تیم است. بازی در نقشه‌ای که از بلوک (خانه)های ۶ ضلعی تشکیل شده است، انجام می‌شود. در ابتدای بازی، تیم چند سلول در اختیار دارد که در بلوک‌های نقشه قرار دارند. در ادامه این سلول‌ها با دریافت دستوراتی که از کد تیم شرکت کننده دریافت می‌کنند 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، جمع آوری منبع و تقسیم سلول است. البته دو عمل جمع آوری 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. 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 می‌گذارند و سلول نمی‌تواند به بلوکی خیلی بلند تر 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.
 - خانه منبع^۳: خانه‌ای 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 می‌کند.

¹ Turn-based

² None

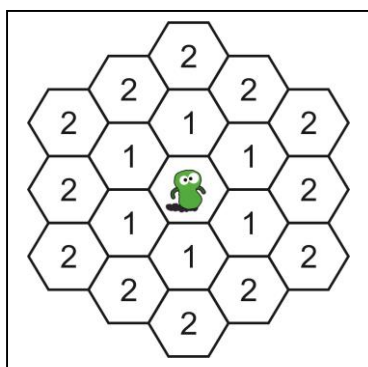
³ Resource

- خانه تقسیم^۴: خانه ای که عمل تقسیم (دو تا شدن) می‌تواند 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 می‌رود.
- خانه غیر قابل عبور^۵: خانه‌هایی هستند که رفتن به آن‌ها ناممکن است.
- خانه معمولی^۶: خانه‌های معمولی نقشه هستند که ویژگی‌های هیچ کدام از انواع فوق را ندارند.

سلول‌ها

هر تیم دارای تعدادی سلول است که در طول بازی سعی می‌کند مجموع انرژی آن‌ها را بیشینه کند. در این مستند همیشه منظور از سلول موجودات سلولی است و اشاره به خانه‌های نقشه ندارد. در ادامه ویژگی‌های این سلول‌ها را توضیح می‌دهیم.

- هر سلول دارای سطح انرژی است که میزان گلوکز ذخیره شده در سلول را نشان می‌دهد، این مقدار حداکثر برابر ۱۰۰ واحد خواهد بود.
- سلول‌ها ضریب جذب^۷ مشخصی دارند که با آن سرعت به جذب منابع از خانه منبع می‌پردازند. اگر میزان منبع موجود در خانه ی کمتر از ضریب جذب باشد، سلول کل منبع موجود را جذب می‌کند.
- شعاع دید هر سلول برابر ۲ است به این معنی که می‌تواند خانه‌هایی را که با فاصله حداکثر ۲ از آن قرار دارند را ببیند. سلول می‌تواند نوع، ارتفاع و حضور یا عدم حضور یک سلول در بلوک‌هایی که می‌بیند را تشخیص دهد. در شکل روبرو می‌توانید خانه‌هایی که یک سلول می‌بیند و فاصله ی آن‌ها تا سلول را مشاهده کنید.



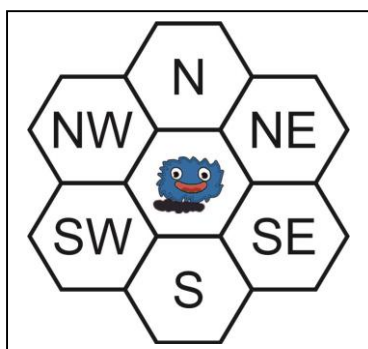
در هر نوبت شما می‌توانید به هر سلول یک دستور دهید و آن سلول در صورتی که شرایط لازم برای اجرای دستور وجود داشته باشد آن را انجام می‌دهد. در صورتی که چند دستور به یک سلول داده شود دستور آخر اجرا خواهد شد. دستورات ممکن در ادامه توضیح داده می‌شوند.

⁴ Mitosis

⁵ Impassable

⁶ Normal

⁷ Gain Rate



- حرکت^۸: دستور حرکت نیاز به یک جهت دارد که مشخص کننده ی یک خانه ی همسایه ی خانه 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.

- حرکت در صورتی انجام می‌شود که بلوک مقصد از نوع غیر قابل عبور نباشد، خالی باشد و ارتفاع آن حداکثر ۲ واحد 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 خانه ی مقصد قرار گرفته و سلول دیگر حرکتی انجام نمی‌دهد.

- تقسیم^{۱۵}: با ارسال این دستور در صورت وجود شرایط زیر سلولی که دستور را دریافت کرده به دو سلول تبدیل می‌شود.

- 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.

⁸ Move

⁹ North

¹⁰ North-East

¹¹ South-East

¹² South

¹³ South-West

¹⁴ North-West

¹⁵ Mitosis

- 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 ۱۵ باشد مقدار انرژی که جمع آوری می‌شود برابر ۱۵ است و در غیر این صورت به اندازه ی انرژی موجود در خانه ی منبع 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 بزنید و موارد ۳ و ۴ فقط در صورتی لازم هستند که بخواهید با C++ کد بزنید. 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 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 getMapSize ()	MapSize getMapSize ()	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

: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

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 getHeight ()	MapSize["height"]
int getWidth ()	int getWidth ()	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(Position)	at(pos)
Block at(int x, int y)	Block* at(int x, int y)	

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
int getTeamId()	int getTeamId()	Cell.team_id

:Energy

دسترسی به انرژی فعلی سلول

Java	C++	Python
int getEnergy()	int getEnergy()	Cell.energy

: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
<code>void gainResource()</code>	<code>void gainResource()</code>	<code>gain_resource()</code>

:mitosis

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

Java	C++	Python
<code>void mitosis()</code>	<code>void mitosis()</code>	<code>mitosis()</code>

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
<code>Direction.NORTH</code>	<code>Direction::NORTH</code>	<code>Constants.Direction.NORTH</code>
<code>Direction.NORTH_EAST</code>	<code>Direction::NORTH_EAST</code>	<code>Constants.Direction.NORTH_EAST</code>
<code>Direction.SOUTH_EAST</code>	<code>Direction::SOUTH_EAST</code>	<code>Constants.Direction.SOUTH_EAST</code>
<code>Direction.SOUTH</code>	<code>Direction::SOUTH</code>	<code>Constants.Direction.SOUTH</code>
<code>Direction.SOUTH_WEST</code>	<code>Direction::SOUTH_WEST</code>	<code>Constants.Direction.SOUTH_WEST</code>
<code>Direction.NORTH_WEST</code>	<code>Direction::NORTH_WEST</code>	<code>Constants.Direction.NORTH_WEST</code>

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 موقعیت¹⁷ خانه ی همسایه یک موقعیت که در جهت ورودی داده شده است.

Java	C++	Python ¹⁸
<code>Position getNextPos (Direction)</code>	<code>Position getNextPos (Direction)</code>	<code>get_next_pos (direction , position)</code>

:X

دسترسی به مختصات X یک موقعیت.

Java	C++	Python
<code>int getX()</code>	<code>int getX()</code>	<code>position["x"]</code>

¹⁷ Position

¹⁸ این تابع برای Python در کلاس Map است.

:Y

دسترسی به مختصات y یک موقعیت.

Java	C++	Python
<code>int getY()</code>	<code>int getY()</code>	<code>position["y"]</code>

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
<code>Position getPos()</code>	<code>Position getPos()</code>	<code>Block.pos</code>

:Height

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

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>

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
<code>String BLOCK_TYPE_NONE</code>	<code>std::string BLOCK_TYPE_NONE</code>	<code>Constants.BLOCK_TYPE_NONE</code>

: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 می تواند داشته باشد که برابر ۱۰۰ است.

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

:Gain Rate

مقدار منبعی که یک سلول می تواند 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

اندازه ی میدان دید سلول که برابر ۲ است.

Java	C++	Python
int	int	Constants.
CELL_DEPTH_OF_FIELD	CELL_DEPTH_OF_FIELD	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

- بازی مرحله اینترنتی جهت آشنایی با نحوه مدیریت و حرکت دادن چند سلول در یک نقشه است. نتایج این مرحله برای انتخاب تیم‌های صعود کننده به مرحله حضوری و رتبه بندی برای انتخاب گروه‌های راه یابنده به مرحله حضوری است.
- بازی مرحله حضوری به مراتب متفاوت با این بازی خواهد بود و به صورت رقابتی است، به این معنی که هر بازی بین چند تیم برگزار خواهد شد.
- ممکن است نسخه جدیدی از سرور و کلاینت، در دسترس شما قرار گیرد. حتما سایت مسابقه و ایمیل خود را چک کنید. بروزرسانی سرور و کلاینت باعث ناسازگاری و تغییر در کدهایی که شما نوشته‌اید نخواهد شد.
- در صورت داشتن سوال یا بروز هر گونه مشکل به آدرس <http://javachallenge.ir/qa/> مراجعه کنید و سوال خود را در آن صفحه مطرح کنید.
- تغییر هر بخش کد غیر از کلاس AI با یا بدون استفاده از reflection منجر به حذف تیم خواهد شد.
- کلاس‌ها و توابعی که در طول مستند بررسی شدند بیشتر مورد نیاز هستند، ولی برای آشنایی بیشتر با نحوه عملکرد می‌توانید به خود کد مراجعه کنید.
- نحوه‌ی ارسال پاسخ‌ها از طریق سامانه‌ی داور شریف (sharifjudge) خواهد بود که اطلاعات تکمیلی در این مورد متعاقبا اعلام خواهد شد.

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

پس از اجرای JavaChallenge2015 یک محیط گرافیکی مشاهده می‌کنید، در این قسمت روی آیکون 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 می‌ماند تا شما کد خود را Run کنید. نحوه‌ی شروع به نوشتن کد و اجرای client در زبان‌های برنامه نویسی مختلف در ضمیمه‌های بعدی توضیح داده می‌شود. (کد خود را به صورت مجزا Run کنید و نگران ارتباط آن با Server نباشید، این قسمت از کار را ما به عهده می‌گیریم).

پس از Run کردن کد خود، روی دکمه‌ی Start Game کلیک کنید و از بازی لذت ببرید.

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 کلاس، تابع و متغیرهای لازم خود را در فایل AI.java و پکیج client 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.

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 کلاس‌ها، توابع و متغیرهای لازم خود را در فایل AI.cpp یا فایل‌های جدیدی که خودتان می‌سازید اضافه کنید.

نحوه اجرا

برای نصب client مربوط به C++ در Linux ابتدا باید package‌های زیر را با استفاده از وارد کردن آن‌ها 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 فولدری که در workspace آن‌زپ کردید را انتخاب کنید فولدر (JC) و گزینه Finish را بزنید.

حال می‌توانید project ایجاد شده را compile و اجرا کنید.

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

(۱) درست کردن فایل Makefile برای پروژه و فرستادن آن به همراه کدها

(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.

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 کلاس، تابع و متغیرهای لازم خود را در فایل `AI.py` یا فایل‌های جدیدی که خودتان می‌سازید 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 روش‌های 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، می‌توانید 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 `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 `start` را در `JavaChallenge2015` بزنید.

ضمیمه ۵: اضافه کردن JDK 1.8 به Eclipse

توجه کنید که اگر از eclipse نسخه^{۱۹} 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.

