

AI Challenge Information

Optional: If you are putting the AI Challenge on your own machine, get it from: http://ai-contest.com/starter_packages/java_starter_package.zip

Using the Tools (adapted from http://ai-contest.com/using_the_tools.php, http://ai-contest.com/starting_your_own.php)

Watch Two Bots Fight

Now it's time to watch two bots fight. For this step, we will have to use a command-line terminal. But don't worry, it's really easy! To open a command-line terminal in Windows, open the "run" dialog, type "cmd", then press the Run button. You should get a black window with a flashing white cursor.

First, go to the starter package.

Next, we will invoke the game visualizer. Don't worry if you don't understand this long command. All you have to do is copy and paste it.

```
java -jar tools/PlayGame.jar maps/map7.txt 1000 1000  
log.txt "java -jar example_bots/RandomBot.jar" "java -jar  
example_bots/DualBot.jar" | java -jar tools/ShowGame.jar
```

The above command plays a game between two of the sample strategies. Once the game is complete, the visualizer window pops up to play back the game graphically. Hit the play button to watch the game.

In the playback you just saw, player 1 is called RandomBot, and player 2 is called DualBot. RandomBot basically works by choosing random legal moves. It's not very smart. DualBot uses a slightly more intelligent strategy, and changes its level of aggressiveness depending on whether it is winning or losing. There are other sample strategies that come with the starter package:

- BullyBot
- DualBot
- ProspectorBot
- RageBot
- RandomBot
-

You can also choose from a large variety of maps. The starter packages each come with 100 different maps. For example, to watch RageBot play against ProspectorBot on map number 43, use the following command.

```
java -jar tools/PlayGame.jar maps/map43.txt 1000 1000
log.txt "java -jar example_bots/RageBot.jar" "java -jar
example_bots/ProspectorBot.jar" | java -jar
tools/ShowGame.jar
```

Play around with different combinations of bots on different maps. Have fun laughing at how much RandomBot and RageBot suck. In the next tutorial, you will create a simple bot of your own and watch it play against the sample bots. You can even play your strategy against itself!

Note for people using Linux: if you are SSH-ed into some server other than your local machine, then you must use X-forwarding in order to see the graphical playback interface. Alternatively, you can just follow this tutorial on your own local machine to solve this problem.

Opening a Command Prompt

In this tutorial, you will use a command prompt (also known as a terminal) to issue commands. Don't worry, it's really easy! If you're using a Windows machine, open the Run dialog, type "cmd", then click the Run button. A black window containing white text and a flashing cursor should pop up.

Once you have a command prompt open, you also have to point it at the location where you unzipped the Java starter package.

Compiling the Java Starter Package

The Java starter package contains a simple working bot that you can use as a starting point. To compile the starter package, use the following command. Remember that you must have the terminal pointed at the correct directory, otherwise the Java compiler won't know where to find the code files.

```
javac *.java
```

You'll know that the Java code files got compiled correctly if you see the file *MyBot.class* was created.

Watching Your Bot Play Against Itself

If the compile goes smoothly, then you can watch your bot play against itself using the following command.

```
java -jar tools/PlayGame.jar maps/map7.txt 1000 1000
log.txt "java MyBot" "java MyBot" | java -jar
tools/ShowGame.jar
```

You can also watch your bot play against one of the sample strategy using the following command.

```
java -jar tools/PlayGame.jar maps/map7.txt 1000 1000
log.txt "java MyBot" "java -jar example_bots/RandomBot.jar"
| java -jar tools/ShowGame.jar
```

Make an Easy Improvement to Your Bot

Open the file `MyBot.java`. This is the code file that you will be working on to improve your bot's strategy. Inside, you will see that there is some sample code already in this file. Right near the top of the file, you'll see the following lines of code.

```
// (1) If we current have a fleet in flight, just do
nothing.
if (pw.MyFleets().size() >= 1) {
    return;
}
```

These lines are what stop your bot from sending all of its ships at once. The number that appears in this code is the maximum number of fleets that your bot can have in flight at any given time. By increasing this number, you can make your bot more aggressive. Change the number from 1 to 2, so the code looks like this:

```
// (1) If we current have a fleet in flight, just do
nothing.
if (pw.MyFleets().size() >= 2) {
    return;
}
```

Save the file and close it. Back in the terminal, recompile your code using the following command.

```
javac *.java
```

Assuming the compile goes smoothly, run your bot against itself again. Use the following command.

```
java -jar tools/PlayGame.jar maps/map7.txt 1000 1000
log.txt "java MyBot" "java MyBot" | java -jar
tools/ShowGame.jar
```

Notice that your bot now generally keeps two fleets in the air at once, instead of only one. You have made your bot more aggressive. You are now an AI programmer. Cool, huh? Show your friends!

You can experiment with different numbers. Try 3 or 4 and see how it does. You can even experiment with keeping different versions of your bot, so that you can play them against each other to see which is better. Try experimenting on different maps, too. If you're really loving tinkering with the code and making simple improvements, check out the [Simple Strategy Guide](#) to see a whole series of small incremental improvements that can be made to the starter package in order to climb up the rankings.