Planet Wars: A Tutorial

Kevin Patel

September 2, 2015

1 Introduction

The aim of this tutorial is to get you started on the Planet Wars AI competition. As mentioned earlier, the competition was originally organized by University of Waterloo, and you can find the archived site at `http://planetwars.aichallenge.org/`. You may find the Tutorials and Strategy Guides helpful. You may eventually have to look at the Planet Wars Specification. Quoting the specification verbatim, the following is the overview of the game:

Overview

Planet Wars is a game based on Galcon, but is designed to be a simpler target for bots. The contest version of the game is for two players.

A game of Planet Wars takes place on a map which contains several planets, each of which has some number of ships on it. Each planet may have a different number of ships. The planets may belong to one of three different owners: you, your opponent, or neutral. The game has a certain maximum number of turns. At the time of this writing, the maximum number of turns on the official server is 200, but it is not yet a part of the specification. Provided that neither player performs an invalid action, the player with the most ships at the end of the game wins. The game may also end earlier if one of the players loses all his ships, in which case the player that has ships remaining wins instantly. If both players have the same number of ships when the game ends, its a draw.

On each turn, the player may choose to send fleets of ships from any planet he owns to any other planet on the map. He may send as many fleets as he wishes on a single turn as long as he has enough ships to supply them. After sending fleets, each planet owned by a player (not owned by neutral) will increase the forces there according to that planets growth rate. Different planets have different growth rates. The fleets will then take some number of turns to reach their destination planets, where they will then fight any opposing forces there and, if they win, take ownership of the planet. Fleets cannot be redirected during travel. Players may continue to send more fleets on later turns even while older fleets are in transit.

2 Example Bots: Basic Strategies, Codes and Matches

All you have to do is fill up your logic in the function 'DoTurn'. You can issue orders using the 'IssueOrder' function. For example, to send 10 ships from planet 3 to planet 8, you would say IssueOrder(3, 8, 10).

2.1 Do_Nothing Bot

Lets get started with a basic **Do_Nothing** bot. As the name suggests, it does nothing.

1. Download the code from `https://www.cse.iitb.ac.in/~kevin.patel/AI_temp/src/Do_Nothing.tar.gz`
2. Extract the files
3. Run make

```
make
```

4. Rename the executable file *MyBot* to *do_nothing*

5. Run the following command

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

You can also view the match at `https://www.cse.iitb.ac.in/~kevin.patel/AI_temp/index.php?game_id=1` As none of the bots are doing anything, the game will continue till the number of moves specified (1000), and result in a draw. Now lets make them do something.

2.2 Capture_Neutrals_Then_Sit_Tight Bot

This bot will just capture all neutrals, then do nothing. In each move, it does the following

1. If a previous order has not reached destination yet, return
2. Selects the neutral planet with the smallest number of ships as the target planet.
3. From among the planets it owns, selects the one with the largest number of ships as the source planet.
4. Sends half number of ships of source planet to target planet.

Lets try pitching it against the **Do_Nothing** bot.

1. Download the code from `https://www.cse.iitb.ac.in/~kevin.patel/AI_temp/src/Capture_Neutrals_Then_Sit_Tight.tar.gz`
2. Extract the files
3. Run make

```
make
```

4. Rename the executable file *MyBot* to *capture_neutrals_then_sit_tight*
5. Run the following command

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

You can also view the match at `https://www.cse.iitb.ac.in/~kevin.patel/AI_temp/index.php?game_id=2` As was mentioned in the overview, the player with larger number of ships wins. This bot spends existing ships in the beginning to win neutral planets, which leads to increase in it's overall *growth rate*. By the time the game ends, it has a significantly larger number of ships than the **Do_Nothing** bot.

2.3 Capture_Neutrals_Then_Enemies Bot

A slight modification to the previous one, in each move, this bot does the following:

1. If a previous order has not reached destination yet, return

2. 2*a*) If not all neutral planets captured, select the neutral planet with the smallest number of ships as the target planet.

2*b*) Else, select the enemy planet with the smallest number of ships as the target planet.

3. From among the planets it owns, selects the one with the largest number of ships as the source planet.

4. Sends half number of ships of source planet to target planet.

Lets try pitching against the previous two bots.

1. Download the code from `https://www.cse.iitb.ac.in/~kevin.patel/AI_temp/src/Capture_Neutrals_Then_Enemies.tar.gz`
2. Extract the files
3. Run make

   ```
   make
   ```

4. Rename the executable file *MyBot* to *capture_neutrals_then_enemies*
5. Run the following command

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

   where ABC could be either of the above two bots

This bot does a proper battle with the enemy, once all the neutral planets have been captured. This strategy is often known as *boom* in gaming community. However, one can easily see that such an approach can take longer to win. Given the limit of number of turns, this may not be a good strategy. Check out the following matches.

- With 1000 moves limit, this bot looses - `https://www.cse.iitb.ac.in/~kevin.patel/AI_temp/index.php?game_id=4`
- With 2500 moves limit, this bot wins - `https://www.cse.iitb.ac.in/~kevin.patel/AI_temp/index.php?game_id=5`

2.4 Capture_Neutrals_Then_Enemies_Impatient Bot

All the previous bots, after giving a fleet an order to move, would wait for that fleet to reach it's destination. Instead one can have an impatient bot, which does not wait. The strategy is as follows:

1. 1*a*) If not all neutral planets captured, select the neutral planet with the smallest number of ships as the target planet.

1*b*) Else, select the enemy planet with the smallest number of ships as the target planet.

2. From among the planets it owns, selects the one with the largest number of ships as the source planet.

3. Sends half number of ships of source planet to target planet.

Lets try pitching against the previous three bots.

1. Download the code from `https://www.cse.iitb.ac.in/~kevin.patel/AI_temp/src/Capture_Neutrals_Then_Enemies_Impatient.tar.gz`
2. Extract the files
3. Run make

   ```
   make
   ```

4. Rename the executable file *MyBot* to *capture_neutrals_then_enemies_impatient*
5. Run the following command

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

where ABC could be either of the above three bots

And this is where the variation starts. Depending on the map (*i.e.*, planets position, growth rates, etc.) the performance of this bot varies. Check out these matches

- `https://www.cse.iitb.ac.in/~kevin.patel/AI_temp/index.php?game_id=6`
- `https://www.cse.iitb.ac.in/~kevin.patel/AI_temp/index.php?game_id=7`
- `https://www.cse.iitb.ac.in/~kevin.patel/AI_temp/index.php?game_id=8`

2.5 Default Bot

Finally, the default starter package contains a bot which does the following:

1. If a previous order has not reached destination yet, return
2. Selects the planet (either neutral or enemy) with the smallest number of ships as the target planet.
3. From among the planets it owns, selects the one with the largest number of ships as the source planet.
4. Sends half number of ships of source planet to target planet.

Lets try pitching against the previous two bots.

1. Download the code from `https://www.cse.iitb.ac.in/~kevin.patel/AI_temp/src/default.tar.gz`
2. Extract the files
3. Run make

   ```
   make
   ```

4. Rename the executable file *MyBot* to *capture_neutrals_then_enemies_impatient*
5. Run the following command

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

   where ABC could be any bot

Check out these matches.

- `https://www.cse.iitb.ac.in/~kevin.patel/AI_temp/index.php?game_id=9`
- `https://www.cse.iitb.ac.in/~kevin.patel/AI_temp/index.php?game_id=10`
- `https://www.cse.iitb.ac.in/~kevin.patel/AI_temp/index.php?game_id=11`
- `https://www.cse.iitb.ac.in/~kevin.patel/AI_temp/index.php?game_id=12`
- `https://www.cse.iitb.ac.in/~kevin.patel/AI_temp/index.php?game_id=13`
- `https://www.cse.iitb.ac.in/~kevin.patel/AI_temp/index.php?game_id=14`

The matches against the impatient version definitely gives a really important observation - Ships, while traveling, are not as useful as they are when waiting on a planet. Try to make use of this fact.

Conclusion

Start with these basic version, and see if you can develop more complex ideas. For instance, all these bots were sending ships from one planet and to one planet only. But more than one order can be issued per move. We shall soon put up the live site where you can submit your code, and also put up the evaluation scheme. But we advice you to get started with the coding, and compare with these bots as well as your colleagues locally if needed. In case of any bugs in the document, notify me via email.