CS 4633/6633 Programming Assignment 3: RISK

Chris Archibald

November 5, 2014

This programming assignment is due on Tuesday, December 2 2014, at 11:59pm.

Honor code note: For this assignment you will be in teams of 2 or 3. You are welcome to share all code writing with your teammates, and share support code between groups. Any copying of code from outside sources (outside this class) will be considered an honor code violation.

This assignment involves playing the game of RISK. You should first download the `PA3.zip` file from Blackboard and unzip in on your computer. The provided code has been tested with Python 2.7. It might work with changes on other version of Python, but you are on your own for that.

Required: You will need to make sure the Python Imaging Library (PIL) is installed on your system. This can be found at

`http://www.pythonware.com/products/pil/`

The folder you are provided contains:

- `PA3_Handout.pdf` - This file that you are reading

- `README` - Gives instructions on how to run the various tools and programs.

- `matches` - This is the folder where matches that your ai participates in are stored.

- `ai` - This is the folder where the ai files are stored. It is provided with several following ai files. Those files with `_ai` are ais I wrote that are compatible with all of the provided tools. Those files with `_bot` were provided with the code that the GUI was based on, and only works with the `risk.pyw` GUI. Additionally, no details are provided about how they work.

    - `random_ai.py` - This is an agent that gets all of the available actions and picks one at random.

    - `attacker_ai.py` - This is an agent that places all troops on territories bordering the opponents and always attacks if that is an option.

    - `donothing_ai.py` - This is an agent that only places troops randomly on its territories, but never attacks. A good opponent for early testing.

- – `heuristic_ai.py` - This is an agent that evaluates each possible action using a heuristic function. Currently the heuristic is not implemented, and always returns 0. An easy way to make an AI is to implement this heuristic function.

  – `borg_bot.py`

  – `erasor_bot.py`

  – `Piranha_bot.py`

  – `wyrm_bot.py`

- `world.zip` - This file contains all of the map, territory and other files necessary to construct the board and display the game properly.

- `risk.pyw` - This is the interactive game program. Once you run it, you create players, which can be AI's or human (you). Then start a game. Everything is done by clicking on the map. This will allow you to play against your AI. You can also play your AI against one of the provided AI's, although games played using this program are not saved out to a logfile and thus cannot be replayed and watched. To run this program, type:

  ```
  python risk.pyw
  ```

  from the command line.

- `play_risk_ai.py` - This is a script that will automatically perform a match of specified length between the specified AI's and save the game log file. To run the code you should type:

  ```
  python play_risk_ai.py AI_1.py AI_1_Name AI_2.py AI_2_Name ...num_games
  ```

  from the command line. Every player should be given a different name, but the same AI can be submitted for multiple players (with different names.) This will automatically play all the games and save the results. It will rotate the players order between the different games.

- `risk_game_viewer.py` - This is the script that allows you to view games played using the `play_risk_ai.py` script. It just needs to be passed in the logfile name. Run it by typing:

  ```
  python risk_game_viewer.py LOG_FILE
  ```

  from the command line.

- `aihelper.py, riskengine.py, riskgui.py, risknetwork.py, risktools.py, turbohelper.py` - These files provide the support for the game, and shouldn't be edited. `risktools.py` was written by Dr. Archibald, and contains all of the datastructures your AI might need to reference and use. You should definitely look at it as a reference, but do not edit it or you could break the whole operation.

Rules

The rules of the class competition. We will be playing one-on-one RISK, following the rules of RISK as outlined in the rules at

http://www.hasbro.com/common/instruct/risk.pdf

Ignore the special 2-player rules they propose, with buffer armies. We will just play 2 player game, but with the normal world domination rules.

Your agent will be presented a state of the game and must return a valid action. Each agent will have a **10 minute time-limit per game** in which to make its decisions. Your agent will be passed the time remaining for each turn as well. You are free to use this time as you see fit, but once you run out of time, the game is over.

The games will be capped at **5000 actions**. If the game is not over by this point it will be considered a tie. This limit might be changed later if we find that it is too small to allow for games to normally finish.

A match between two agents will consist of 10 games being played, five with each player going first. The agent with the most wins out of 10 will be considered the winner. In case of a tie, sudden-death play of games will be played with a random player going first.

Part 1 [100 points]

The main goal of this assignment is for you to write a competitive AI for the game of RISK.

Beating the random agent [80 points]

You will pass of this assignment if your AI can beat the random_ai.py, in a ten game match. To demonstrate this you will submit the code for your AI (your .py file) as well as logfiles of the match in which your agent beats the random_ai.py.

Beating the attacking agent [20 points]

You will pass off this assignment if your AI can beat the attacker_ai.py, in a ten game match. To demonstrate this you will submit the code for your AI (your .py file) as well as a logfile of a match in which your agent beats the attacker_ai.py. Please make sure all of your relevant code is included in your ai file, since Dr. Archibald will be running a tournament between all the submitted AIs.

Part 2 [50 points]

This portion of the assignment requires that you provide a written description of how your agent works and why you designed it like that. This should be around 2 pages minimum, but can be more if you have more details to share.

1   Extra credit [50 points]

Your submitted agent will compete against all of the other AIs from the class in a class competition. Extra Credit points will be given out for performance in this competition.