

# 1 Code ReadMe

The variant of blackjack that is used as the environment for this learning problem is described by

<https://hadovanhasselt.files.wordpress.com/2016/01/easy21.pdf>. Any non-standard libraries used are listed within section 1.2.

Section 1.1 describes the structure of the environment directory, with a short summary of the responsibility of each module. For more detailed notes on each method, please reference the in code comments.

## 1.1 Directory Structure

The hierarchy of its contents are described as follows:

1. **RL\_PolicyIteration** > Top level directory containing the solution.
  - (a) **Check.py**: Script containing the methods required for testing the Game class with an executable for running the test.
  - (b) **Game.py**: Module with Class containing all fields and methods required by the Blackjack Easy21 game environment.  
**Methods:**
    - i. **draw**: Method that simulates drawing cards from the deck.
    - ii. **step**: Method that uses draw and simulates making a move in the game.
  - (c) **LinFuncApprox.py**: Script containing the method for linear function approximation that utilises the Game class and executable for running and producing output.
  - (d) **Logging.py**: For housing any generic logging methods that would be required across the solution.
    - i. **func timer**: Method providing a function handle for timing the various RL algorithm functions.
  - (e) **MonteCarlo.py**: Script containing the method for Monte Carlo control that utilises the Game class and executable for running and producing output.
  - (f) **Sarsa.py**: Script containing the method for Sarsa( $\lambda$ ) that utilises the Game class and executable for running and producing output.
  - (g) **Setup.py**: Module with methods for dictionary initialisation and plotting that are used across the scripts detailed above. Also, contains all the package import statements required by the solution.
  - (h) **output** > Directory containing all output files. The output results files are space separated text files, and the output plots are .png files, stored in the plots sub folder.

## 1.2 Framework

The non standard packages installed in addition to the basic python distribution are summarised as follows:

1. **Language: Python 2.7.8:**
2. **Additional package dependencies:**

- (a) numpy 1.11.0
- (b) matplotlib 1.5.1

## 1.3 How to run?

To run the methods and also produce the required output, simply execute the responsible script as detailed within section 1.1.

e.g. To run Monte Carlo control simply call 'MonteCarlo.py' from the console, from within the directory path. The output files will automatically be created within the output directory.

For each of the executable scripts, the progress of the RL algorithm is provided within the console output in the form of the episode number (and mean square error and  $\lambda$  value when required).