CMPSCI 687: Homework 3
Fall 2020

Philip S. Thomas
University of Massachusetts
pthomas@cs.umass.edu

For this part of the assignment, you will implement value iteration (modified to terminate when the value function estimate has not changed significantly between two iterations). Your program will read an MDP from a file, run value iteration on the MDP, and output the final estimate of the optimal value function and the policies that are greedy with respect to this value function. As a soft introduction to C++, we are providing you with most of the code here: your job is to fill in the missing lines in the function `valueIteration`, marked with a comment saying "TODO". Do not change the code logic outside of the valueIteration function (you may add new functions if you like, but do not modify any of the other functions in your final submission or it may fail to run as expected in our auto-grader). You may introduce additional include statements, but should only use common libraries (all of the C++ standard libraries are allowed).

You are free to use any IDE or toolchain you would like to program in C++. This assignment is another chance to familiarize yourself with C++. Please look over all of the provided code, and feel free to ask during office hours if you have questions about what some portion of the code is doing. Also, take this opportunity to familiarize yourself with the debugger in your IDE—developing simple programs in C++ is a breeze when you are familiar with how to use the different capabilities of your debugger.

We have provided you (within the provided code) with 687Gridworld.txt, a text file containing the MDP we have been using in class. We will evaluate your program on other MDPs that we are not providing to you. You are welcome to create your own test MDPs, but do not share these with others.

You must submit your main.cpp file. A correct implementation is worth full credit. Partial credit will be given based on whether your implementation produces the correct results on the provided 687-Gridworld and whether it produces the correct results on our other (not provided) test inputs.

1 Due Date

The assignment is officially assigned on October 1, and due on October 8. All students may have an extension until October 15, without penalty, as described for HW1 via email.

2 Eigen

The provided code relies on the Eigen library, which can be found here. To install this library, download the latest release. Inside of the main directory is a directory called “Eigen”. This is the only directory you need. Copy it into your project and ensure that it is in the include path. I usually put a directory called “lib” inside of my project, and place the “Eigen” directory inside of “lib”. I then add “lib” to the list of include directories (how this is done was covered in the C++ tutorial we provided).

3 Compilers

Your code must compile with either gcc (any recent version) or the compiler included with Microsoft Visual Studio. Note that Macs often pretend to be using gcc when they are really using CLang (which we will *not* be using in our grading pipeline).

4 Cheating

For this assignment, all code that you submit must either be code that you wrote, or code that you were provided with the assignment. Although you can discuss high-level topics, all coding must be done individually. You may not use any additional libraries (other than C++ standard libraries and Eigen).

5 Hacking

This is not a security course. If the code that you submit attempts to compromise the machine it is running on (e.g., deleting or reading files outside of the project directory, downloading viruses or back-doors, etc.) it will be reported to the police (and depending on whether the machine was ever used for our DARPA or Army research, the FBI), and will result in your failing the course (via the “inappropriate behavior” clause in the syllabus).

6 Extra Credit

If there are any bugs in the assignment, the first person to point the bug out to me via email (pthomas@cs.umass.edu) will receive 5% extra credit on this assignment.

7 Assignment Changes

If any changes are made to the assignment after it is posted, this document will be updated and a description of changes included below.

- On 10/1/2020 around 8pm, I changed the "cheating" section to not reference BBO algorithms from the last assignment.

- On 10/2/2020 around 11:15am, I changed the main.cpp file to point to the input and output directories as though they are in the directory where the executable is run. This should better align with most project setups.