

The final project of this course will be to implement Q-Learning and Monte-Carlo to learn an optimal policy for the game of Blackjack, and compare the results of both methods.

- You may use the GUI given to you for Assignment 5 as a basis for the project code
- The project must be implemented using JS/HTML like all other assignments in the course
- You must write a new user interface which visualizes the blackjack policy as it is learned over time
 - Like the 'single iteration' functionality from A5 you must have a button which plays a hand (episode) of Blackjack and then updates the values / policy for one episode
 - You should also have a 'toggle iteration' button like A5 which does a specific number of episodes per frame.
 - You can copy all of this code from A5 and fit it to your needs. You basically just change the path-finding logic to be the rules of blackjack
- The policy visualization must look something like this
- The project should be programming in JS/HTML like Assignment 5 to show the improvements to the policy over time as it learns.
- You must implement the rules of Blackjack as follows:
 - You may assume an infinite deck of cards to deal from
 - You must implement the actions: Hit, Stand, Double Down
 - You must properly implement Ace being 11 or 1, depending on the situation
 - You do not have to implement the actions of Split, or Insurance
 - You will lose marks for any of those features not implemented
 - Your policy must converge to the standard Blackjack optimal policy
- This project must also come with a report which is to be submitted as a single PDF file.

THE PROJECT MUST BE DONE SOLO TO OBTAIN THE BONUS MARKS

Final Project Report Specification

For your final project, you are expected to deliver two main components:

1. The code to run your project, including instructions on how to install any required libraries to run the project, as well as instructions on how to run the project or reproduce the results that are included in your report.
2. A written report, delivered as a single PDF document. The written report of the project should include the following sections:

1. Introduction

- a. A brief introduction to what the project is about
- b. Motivation for why this project is interesting and/or important

2. Problem Description

- a. A high-level overview of the problem you are attempting to solve
- b. A description of the environment and possible states in the environment, including the properties of the environment (see end of slides for Lecture 2)
- c. Some screenshots of sample states of that environment
- d. The actions that are possible from states in the environment

- e. Provide the rules of the game, or the subset of the rules of the game that you ended up implementing in your code
- f. The list of goal states / rewards / objectives of the game

3. Methodology

- a. An overview of the algorithm or method used to solve your problem
- b. Why you chose the algorithm, citing the properties of the environment and why you believed this algorithm was the best choice for this problem
- c. Pseudocode of your algorithm, with English description of what it does
- d. Give a reference to where in your submitted code the actual algorithm is implemented
- e. A list of any improvements and/or optimizations you made to the algorithm, and an explanation of how they helped (ie: running time / memory improvements)
- f. A list of heuristics used for your algorithm (if applicable)
- g. A list of things that you may have tried to do but didn't end up working in time

4. Results & Discussion

- a. A description of the experiments you ran to show that your solution worked, including machine specs, sample inputs, and parameters used (ie: alpha beta settings, etc). Screenshots of solutions.
- b. Screenshots / tables of results of experiments that you ran. For example:
 - i. How many iterations it took for each learning method to converge
 - ii. Screenshots or tables of policies over time
 - iii. A video of the system converging over time
- c. A brief discussion of the results you obtained, whether or not you believe they were intuitive (ie: did running the algorithm for longer produce better results)

5. Conclusion

- a. A brief summary of what you did, and whether or not you were successful
- b. What would you do in the future to make more improvements if you had the time