**AI Project**

**The course project is worth 15% of the final** [**grade**](javascript:void(0))**.**

You may work alone or in groups of two or three. The complexity of your task does not depend on the group size, which means that forming a group of three will reduce your workload. A group should submit one programand one final report and all members of the group will get the same grade on this part. You may discuss ideas and algorithms with other groups, but you cannot share your code.

**Task selection and Project proposal (April 2, 3 points):** First, you need to form a group, select a task for your project.You should submit a one-page proposal. It should describe the selected problem, intended approach to solving it, and methods for evaluating your results. You should clearly explain how you will measure the performance and what level of performance will be considered a success.

[**Final**](javascript:void(0)) **report, codes, and poster submission (April 16, 8 points):** The final report should include the description of your system, [summary](javascript:void(0)) of experimental results, main conclusions, and discussion of any surprising discoveries. The report should be at least 3 pages and at most five pages long. You should schedule a fifteen-minute appointment with the TA and give a [demo](http://www.cs.cmu.edu/%7Eeugene/teach/ai02/projt.html) of your program. You may select one member of your group to give a demo or come all together. Each group must prepare the poster for presentation.

**Submitting your project**

* Submission via Canvas’s Assignment.
  + It is your responsibility to submit these project in a timely fashion.
* All files should be zipped together.
* There should be a source code, a report and a readme file explaining in detail the exact steps to be taken to compile and execute the code files and the title page
* A report that presents the performance evaluation of your solution.
  + The report should be properly formatted (an academic format style, such as ACM or IEEE being preferred) and contain quantitative data along with you analysis of these data.

**Presentation in class (April 17, 19, and 24, 4 points):** The members of the group will get the grade depending on the presentation.

**Each group should have a unique project.**

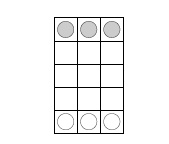
**Some project Ideas**

**Project 1:**

The 3x5 Pawn Game involves two players and six pawns (see the picture). Each player has three pawns, which move in the same way as the chess pawns. The owner of the white pawns makes the first move. A player wins in one of the following cases:

1. One of her/his pawns has reached the last row
2. She has captured all pawns of the opponent
3. All pawns of the opponent are blocked

Implement a program for playing this game. It should prompt a user to specify the player (“white” or ”black”) and then act as this player. For example, if the user enters “white” than the program makes the first move. For the full credit, it should win against the teaching assistant.



**Project 2:**

Implement a program for building decision trees. It should read a file with training and test examples, use the training examples to build a tree, and then classify the text examples. The required output is the classification of the test examples and the tree itself. The input format is as follows:

<classification> <attribute> <attribute> … <attribute>

…

<classification> <attribute> <attribute> … <attribute>

…

<attribute> <attribute> … <attribute>

…

<attribute> <attribute> … <attribute>

The training examples are above the blank line, and the test examples are below. <classification> is either “positive” or “negative, and each <attribute> is a string of lower-case letters. The length of an attribute is at most twenty characters; successive attributes are separated by one or more spaces. For instance, the following file includes three training examples and two test examples

positive fish low good no plate

negative beef high bad plate

negative fish medium good plate

fish high bad noplate

beef medium good plate

**Sample file:**

The file contains six training values and two test values. These values correspond to whether a coin will be valuable to a collector, and the fields correspond to the following table:

|  |  |  |  |
| --- | --- | --- | --- |
| Classification | How many made? | How old is coin? | How much wear is on the coin? |
| Positive | Rare | New | Low |
| Positive | Rare | Old | Low |
| Positive | Common | Old | Low |
| Negative | Rare | Old | High |
| Negative | Common | New | Low |
| Negative | Common | New | High |

**Project 3:**

Consider the problem of finding the shortest path between two points on a plane that has convex polygonal obstacles. This is an idealization of the problem that a robot has to solve to navigate in crowded environment.

**Project 4:**

N-Queens is a puzzle-solving problem involving the placement of *N* chess queens on an *N by N* grid so that no two queens threaten each other. Solutions exist for all natural numbers *n* with the exception of *n* = 2 or *n* = 3. While there are many different ways to represent and solve this problem, the most efficient method is to model it as a constraint satisfaction problem (CSP) using a minimum-conflict heuristic.

**Project 5:**

Object tracking in video.

**Project 6:** Minesweeper Search Algorithm Final Report

Minesweeper is a single puzzle player video games. The objective of the game is to clear a rectangular board containing hidden "mines" without detonating any of them, with help from clues about number of neighboring mines in each field.

**Project 7:**

N-Queens is a puzzle-solving problem involving the placement of *N* chess queens on an *N by N* grid so that no two queens threaten each other. Solutions exist for all natural numbers *n* with the exception of *n* = 2 or *n* = 3. While there are many different ways to represent and solve this problem with artificial neural networks.

**Project 8:**

Forecasting using artificial neural networks.

**Project 9:**

Game: Pirate Ship battle