Computer Science 384
St. George Campus

Thursday, October 13, 2016
University of Toronto

AI Project Assignment
Project Proposals Due: Thursday, November 10 by 11:59 PM
Final Projects Due: Tuesday, December 6, 2016 by 11:59 PM

Late Policy: No late penalties will be applied for this assignment. Materials are due on the date specified unless otherwise stipulated by the instructors. The project report and source code are due on the last day of class (December 6) but if you have remaining grace days, they may be applied. No more than 3 grace days can be applied to any one project, however.

Total Marks: This assignment represents 15% of the course grade.

Teaming: This project is to be performed in groups of 2 or 3. Infrastructure has been set up in Piazza to assist with this. Please refer to the pinned notes of CSC384's Piazza instance for details.

Submission: ** Project team members must each submit identical materials. **

What to submit electronically:

By Thursday, November 10, 11:59 PM

- A pdf copy of your Project Cover Page, as described below. Name your file `csc384-cover.pdf`
- A pdf copy of your Project Proposal, as described below. Name your file `csc384-proposal.pdf`

By Tuesday, December 6, 11:59 PM

- A pdf copy of your Project Report, as described below. Name your file `csc384-report.pdf`
- A zip file containing all the source code for your project together with a `README` file explaining how to run it. Your code must run on CDF unless otherwise discussed and approved in advance of submission. Further details follow. Name your file `csc384-source.zip`

How to Submit: You will submit all components of your project using MarkUs. Your login to MarkUs is your CDF username and password. It is your responsibility to include all necessary files in your project proposal and report submissions. *Warning*: marks will be deducted for incorrect submissions.

Clarification Page: Important corrections (hopefully few or none) and clarifications to the assignment will be posted on the Project Clarification page, linked from the CSC384 web page. You are responsible for monitoring the Project Clarification page:
`http://www.teach.cs.utoronto.ca/~csc384h/fall/Project/Project_faq.html`.

Questions: Questions about the assignment should be asked on Piazza:
`https://piazza.com/utoronto.ca/fall2016/csc384/home`.

1 Introduction

This assignment provides an opportunity for you to be creative and to work on an idea that you formulate. You are to work in teams of size 2 or 3. Individual projects are not allowed except when needed for academic accommodations approved by accessibility services.

The aim of the assignment is for you pick a problem and solve it using one of the problem solving techniques covered in class (i.e., Search, Game Tree Search, CSPs, or Knowledge Representation). Our expectation is that most of you will do so by building on the code that you have developed in assignments, but you may use other software that you find on the web as long as you provide clear attribution and can make it run on teach.cs.

Once you've chosen a problem, you must decide which technique you will use and how. You will then implement a solution (by default, building on software developed during the course), and evaluate your solution. For example, you may want to develop a system for planning routes from one location to another using a search algorithm — perhaps you want to try out some different heuristics and evaluate their relative effectiveness. Maybe you want to use a game tree search algorithm to play a game, developing different heuristics and seeing how effective they are under different timing constraints. Alternatively, you may want to build a CSP to schedule your classes. Perhaps you want to explore CSPs for Sudoku-like puzzles in more detail, by implementing and evaluating different CSP solving strategies like GAC, Forward Checking, or by adding some optimizations atop your CSP to pick "better" solutions from the range of those that satisfy constraints.

In choosing a project topic to explore, you should ask yourself how you will evaluate its success. In a typical evaluation you would design a set of tests or experiments that demonstrate the effectiveness of your solution or that evaluate some hypothesis. For example, an evaluation may show that a heuristic you developed is faster or more informative (results in expansion of fewer nodes of the search space) than another heuristic or compared to blind search. You may wish to try it under different time/space constraints and to see the trade-off of a very informative but computationally expensive heuristic as compared to one that is easy to compute but less informative. You may wish to prove that your heuristic is admissible and if it isn't you may wish to evaluate the quality of the solutions you find. You may evaluate how your solution technique scales as a function of the number of state variables. Alternatively, an evaluation may assess the quality of the answers provided by your system. For example you may wish to perform a user study to assess how many times your CSP planned a garden that correctly reflected your users' constraints, and how many times or under what conditions it was wrong.

Having Trouble Thinking of a Project Topic? If you have difficulty coming up with an idea for a project, we have assembled several ideas that you can use to get started. These are available at the following URL:

`http://www.cdf.utoronto.ca/~csc384h/fall/Project/Project_ideas.html`.

We will add to the list as the semester progresses!

2 The Project Proposal (Due: November 10)

The project proposal will be a *short* document describing your project. Proposals will not be graded, but failure to submit a proposal will affect your overall grade for the assignment. Each project proposal must contain:

Project Cover Page (submitted as `csc384-cover.pdf`):

- Title of the project;
- Names and teach.cs-login IDs of all team members;

Project Proposal (should contain no names) (submitted as `csc384-proposal.pdf`):

- Title of the project;
- Number of team members (2 or 3);
- Type of project: problem solving technique employed (Search, Game Tree Search, CSP, Knowledge Representation);
- Brief project description (250 words or less). This should clearly answer the following questions: *What is your problem? Why is it suited to the problem solving technique you've adopted?*
- Brief evaluation plan (250 words or less). This should clearly answer the following question: *How will you evaluate your solution?* Note that not all solutions may, in fact, work! Clearly illustrating inefficiency in a given implementation can still be a valuable outcome of your work. In cases where you are using data please note the source. In cases where you are developing test cases, please note the nature and number of test cases you anticipate developing.
- Finally, list the roles you expect each team member to play. Note that all team members should have some substantial role in the realization of your project. Each may implement a different game tree search heuristic, for example, or one may encode a problem as a CSP while the other explores propagation techniques.

Example Project Proposal

- Title of Project: Wordoku
- Type of Project: Search
- Number of Team Members: 2
- Project Description: In our project, we will use search to play a simplified version of *Wordoku*. In *Wordoku*, you are given a 3x3 matrix and a list of 9 letters. You have to arrange letters to form valid three letter words in every row and column and across the two diagonals of the grid (from the top to the bottom corners). We will explore the use of uninformed and heuristic search to solve this puzzle. Well recognize goal states using a list of legal three letter words (http://www1.cs.columbia.edu/kathy/cs4701/3.txt) and build heuristics with a list of bigram frequencies (http://www1.cs.columbia.edu/kathy/cs4701/bigram.txt). The bigram frequencies give probabilities for letters, conditioned on preceding letters. We can use these to pick letter arrangements that are more or less likely. This project is suited to heuristic search because its a search over letter combinations. We will start with a random assortment of letters, and use heuristics to replace letters in the matrix. We will know if we arrive at a goal state, because we will be able to see if our final array of letters contains legal words in all directions, as defined by our dictionary.

- Evaluation Plan: We will run the same searches using several different search strategies and compare both the time and the space that they require. More specifically, we'll compare uninformed search strategies (i.e. depth first and breadth first search) to heuristic searches (greedy search, A* search). We'll keep track of the number of letter combinations each search explores and the maximum amount of memory we use during each search. We'll identify situations and examples where our heuristic searches perform either very well or very poorly and provide explanations of these examples in our discussion.
- Roles for Team Members: One team member will be responsible for encoding heuristic functions that are based on bigram frequencies. The other will encode the state space for Wordoku and build tools to check that letter combinations form legal words. Both team members will contribute to the write-up: one will write the game description and methods while the other will write the results and discussion.

3 The Project Report **(Due: December 6) (submitted as** `csc384-project.pdf`**)**

1. Each report must have a **title page** containing the following information:

   - Title of the Project;
   - The names and teach.cs-login IDs of all team members;
   - The roles played by each member of the team (e.g. problem encoding, experimental assessment, manuscript author, ...) and whether they were major or minor roles;
   - Type of Project: the problem solving technique employed, i.e., Search, Game Tree Search, CSP, KR, or Other.

2. Following the title page, include a **report body**. This can be a maximum of 5 single-spaced pages, formatted in 12pt font. Sections for the report body should be as follows:

   - **Project Motivation/Background**. Here, you will describe the problem you are trying to solve or the application you are trying to create. Also describe your approach to the problem (e.g. Search, Game Tree Search, CSP, or KR) and the rationale for choosing this approach.
   - **Methods**. Here, describe the details of your realization. How did you formulate your problem and what algorithms did you employ to solve it? For example, depending on your problem type you may need to describe your state encoding — what are the state variables and domains; what are the successor functions; how you encoded your constraints (and why). If you developed heuristics, describe whether they are admissible or not and any other properties they have. The above are just examples and are not exhaustive.
   - **Evaluation and Results**. Here, describe your evaluation objectives and strategy, and your results. In particular, describe the way you've chosen to evaluate your approach (i.e. how you will determine if your approach works). Evaluation metrics could include the number of nodes expanded in a search algorithm or the amount of time or memory that you used. We encourage the use of diagrams, graphs, and/or tables to summarize experimental results and to convey important points. Note that it's ok if your system proves to be inefficient in some way; that's still a result and we want to know. In addition to graphs and tables, provide a written summary of your findings and their implications, if any.

- **Limitations/Obstacles**. Here, document any obstacles you encountered during your implementation or shortcomings you discovered in your solution approach.
- **Conclusions**. Finally, explain what you learned and how you might improve or modify your program were you to try again in the future. Other reflections are welcome.

3. You may include up to 2 additional pages after the report body for **citations and references** or any other attributions or acknowledgements.

4 The Project Source Code (Due: December 6) (submitted as `csc384-source.zip`)

The realization of your project must run on teach.cs. All source code to run your completed project on CDF must be submitted, together with a `README` file explaining how to run the code, in a single zip file.

5 Marking Scheme

The following items will be among those considered when we mark your project.

- Did each team member contribute? One team member may be responsible for engineering test cases while another may be responsible for implementing a search heuristic. Each role should be significant and clarified in the report.
- How novel or interesting is the problem you chose? Imaginative ways of using AI technology will be rewarded, but this is not a mandatory criterion for a strong grade.
- How appropriate is the solution technique you chose? For example, you will get fewer marks if you chose to use search for a problem that would be better solved using constraint satisfaction.
- Does your solution to the problem together with your discussion demonstrate a deep understanding and mastery of the concepts in the course? Did you choose a good encoding of your problem? Did you demonstrate an understanding of how to exploit properties of the problem to develop an effective state representation, good heuristics, maximally exploit independence in your Bayes Net, etc.?
- Is your evaluation strategy comprehensive and the outcome convincing? If the evaluation strategy or realization had deficits, did you demonstrate that you understand what more you might have done to make this a convincing strategy. Did you do enough to convince us that your solution shows promise, or, if it didn't work as well as you had wanted are you able to explain why in your report?
- Clarity of the report: was the material in the document well structured; was the writing clear; did you explain the problem and your approach clearly and methodically; did you specify the CSP, search space, or game clearly; was the evaluation presented using appropriate graphs and tables (where relevant) and was there a suitable discussion of the findings? Are the conclusions you draw justified by your results?
- How significant are your findings? E.g., have you identified and solved a problem that hasn't been solved before or solved it in a potentially better way. Significance of the work is not expected nor is it a mandatory criterion for a strong grade, but it will be rewarded.

6 Other things to consider

- Tempting as it can be, don't spend all your time building a cool interface rather than working on the AI component of your project.
- Make sure you have the expertise to develop your application. E.g., if you're developing a Bayes Net to do medical diagnosis, make sure you have sufficient medical expertise.
- If you require data, make sure you can access that data or create reasonable synthetic data.
- You have to add value from the starter code that you use. Again, we are assuming most people will build from the software developed in the three previous assignments but if you use code from the web, you must provide some AI-related contribution. This can either be with respect to your problem formulation, or the development of heuristics, or it might be with respect to an extensive problem encoding and evaluation.

HAVE FUN and GOOD LUCK!