



Al Imam Mohammad Ibn Saud Islamic University
College of Computer and Information Sciences
Computer Science Department

CS 340 – Artificial Intelligence
Spring 2020 - Semester Project
Developing Interactive AI Tutorials

Teaching is the best way to learn.

Contents

- 1. Description**
 - 2. Deliverables**
 - 3. Grading**
 - 4. Deadlines**
 - 5. Supplements**
-

1. Description

In this project you will write three interactive AI tutorials (Solving Problems by Searching, Adversarial Search and Constraint Satisfaction). You will write them as Jupyter notebooks which implement and explain some of the essential algorithms you have seen in class.

The Jupyter notebook has to explain the algorithms and related concepts in both Arabic and English, and most importantly it has to explain how to implement the algorithms from scratch.

The notebook should also contain at least one appropriate formulated problem for each notebook, to which the algorithms are applied, and, where applicable, their performance evaluated and compared. These problems should preferably be simplified problems from our background and daily interactions.

You should guide the reader to test and change variables in the problem/algorithm so they can understand the important concepts.

The following table contains the algorithms you need to explain, implement, and show their performance on the formulated problem.



Notebook Title	Main Concepts	Algorithms
Adversarial Search	<ul style="list-style-type: none">• Game trees<ul style="list-style-type: none">○ Game as search (initial state, state space, goal state)○ Evaluation function○ Alpha-cut and Beta-cut○ Performance measure○ Example for each concept○ Any knowledge you learnt about Games	<ul style="list-style-type: none">• Min-Max• Alpha-Beta

A suggested notebook layout is as follows:

- Introduction
 - Explains the objectives and main concepts.
- An example game X-O
 - Explains the problem and its formulation.
- The algorithms, for the two algorithm
 - Explanation
 - Implementation
 - Applying the two algorithm on the X_O game (graphic interface is desirable)
 - Testing the two algorithms on several samples of X_O games (10 times) and reporting the results (number of wins, time, the size of the tree, depth tree).
 - NB: The horizon and the evaluation function should be the same for both algorithms.
 - Compare and Interpret your experimental results
- References and resources

To help the reader visualize the algorithm the notebook should also contain figures, and preferably useful animations.

It is very important to correctly cite any resource you use, failure to do so is considered plagiarism. Refer to section 3. Grading.

Examples of existing notebooks that you may use as guidance, however using the existing search code is not accepted.

1. <https://github.com/norvig/pytudes#pytudes-index-of-jupyter-ipython-notebooks>
2. <https://github.com/Calysto/aima3/blob/master/notebooks/search.ipynb>



2. Deliverables

1. The notebook and any related files like figures, animations, data files... all submitted in a zipped folder.
2. An oral demonstration.

3. Grading

Component	Marks	Notes
Explanation of concepts	5	Correct and clear explanation of concepts using figures.
Problem formulation	3	Appropriate examples, and correct formulation and implementation.
Algorithm Implementation	10	Correct execution, enable reader to adjust variables.
Testing		Testing the algorithm on several samples and reporting the results in a table format.
Demonstration	2	Communication, and team work skills will be assessed.
Understanding of the code	100%	The total grade will be affected by this percentage. Each student will be assessed individually.
Total	20	

Final grade out 15= (3/4) grade out of 20

There are up to 3 bonus marks for outstanding projects.

Any suspicion of cheating and plagiarism will be dealt with seriously and raised to the department which will result in failing the course. Also make sure there is no plagiarism especially in the theoretical background, you need to write the theoretical background in your own words in both Arabic and English.

4. Deadline

The deadline is on Saturday **05/November/2022.**

5. Project discussion : the 11th week, time see schedule section.



6. Supplements

1. How to install Anaconda, it includes Python, several IDEs and python packages.
<https://docs.anaconda.com/anaconda/install/>
2. The basic topics in python that you might need in the project depending on your implementation are dealing with:
 - a. Lists.
 - b. Numpy arrays.
 - c. Reading text files.
 - d. Defining functions.
 - e. Defining classes.
3. Jupyter notebook documentation (contains examples):
<https://jupyter-notebook.readthedocs.io/en/stable/>
4. Introduction to Jupyter notebook:
<https://realpython.com/jupyter-notebook-introduction/>