

King Saud University
College of Computer and Information Sciences
Department of Computer Science

CSC 361: Artificial Intelligence
Second Semester 1442-1443 H (Winter 2021)
Instructor: Yousef Alohal

Project
Due Date: 21 April 2021

1 Introduction

Students can work in groups of 2-3 students. The project is composed of 2 consecutive parts. The first part is to develop a program and write a report. Whereas in the second part a tournament will be hosted.

2 Project description:

In this project you will program an agent that uses Minimax to play the game of Othello. You do not need to implement the game of Othello. You can use the Java implementation included in the project file which is a very simple implementation of the game of Othello. You can test it by running the file "Test.java", which should run a game where both players generate moves at random.

3 Project files description:

The Othello implementation is structured as follows:

- **OthelloMove.java**: this class contains a "move" (which player made the move and the coordinates of the move)
- **OthelloState.java**: this is the core class, which implements most of the functionality of the game. The functions you should be aware of, for implementing minimax are as follows:
 - **public List getValidMoves ()**: returns the list of moves for the next player to move.
 - **public List getValidMoves (int player)**: same as before, but you can specify which player you want to generate moves for.
- **OthelloPlayer.java**: this is an abstract class defining an agent that plays Othello. Your agent should be implemented as a class that extends this one.
- **OthelloRandomPlayer.java**: an example agent that plays Othello, it chooses moves at random.
- **README**: has detailed information on how to run and compile the game

Specifically, what we are asking you to do is the following:

- Create a new class that extends OthelloPlayer (call it something original, to prevent any name clashes with the classes that your classmates will create, since we will put all the classes together at some point, to create a tournament!).
- Reimplement public Square getMove(GameState currentState, Date deadline) method
- Within this new class, implement an agent that plays Othello using the standard minimax algorithm, as we studied it in class.
- As the evaluation function, just use the "score" function that is provided to you in the OthelloState class (make sure that your bot can play both as the first or second player).
- To make sure your agent works, make it play against the OthelloRandomPlayer we provide. Your agent should defeat it easily!

4 Alpha-Beta

Implement a second agent, which uses alpha-beta pruning. Compare the times that your two agents take to search up to different depths and write that in your report.

5 Tournament

Implement another agent, which instead of receiving the depth at which to perform search, it receives a certain amount of time (in milliseconds) that it can use to search. Make sure that your bot returns a solution within this time (you can assume that at least you will have 1 Second).

Hint: a good way to do this is by making your agent first search at depth 1. Then, if there is still time, search at depth 2. If there is still time, go for depth 3, etc. Also, make sure that you have code that cancels the search if enough time has passed.

We will collect all the agents that comply with this and play a tournament.

6 Important notes

- 1- All project files for this course must be submitted using LMS. Do not e-mail your submission!
- 2- Your source code (should include the othello implementation, to make it self-contained), written documentation for your program. A MAKE file for your code. MAKE is a utility to help others automatically compile your code. The MAKE file should produce a single standalone executable piece
- 3- In case you change your group, provide me with your groups members' names before 15 March 2021.
- 4- All group members must participate in all parts of the project programming and report.
- 5- The time and date of the tournament will be determined later and announced in class.

7 Collaboration policy:

You are free to form groups to study and discuss tasks, assignments, and projects. However, you must write up your own assignments and code from scratch independently, and you must acknowledge in your submission all the resources. Prepare your tasks by taking into consideration the following points:

- It is not allowed to share your own work with others.
- It is not allowed to look at the works of other students.
- It is not allowed to upload your write-up or code to a public repository (e.g. github) before the end of the semester.