EE 562: Artificial Intelligence for Engineers

Autumn 2023

HW Assignment 3 (Program): 25 pts

Turn In Here

Download the skeleton code here:

Kalah game starter code

Due Monday, Nov 6 at 11:59 pm,

Problem Description

The purpose of this assignment is to try out the game-playing techniques we have studied as well as add your own ideas. The game is Kalah with the following rules:

- Each player has 6 holes and a Kalah as shown in the lecture slides.
- A turn consists of selecting one of your own holes, picking up all the stones, and distributing them in each consecutive hole, including your Kalah but not the opponent's Kalah, in the counterclockwise direction.
- If your last stone lands in your own Kalah, you get an extra turn.
- If your last stone lands in your empty hole, you put all the stones in the opponent's opposite hole in your Kalah. You will also place your last stone in your Kalah. If you are unable to capture any of your opponent's stones, because you landed across from an empty hole, leave your last stone where it is and do not move it to your Kalah.
- If you run out of stones on your side, the opponent takes all the stones left on his side and puts them in his Kalah

Requirements

- You will work alone. Feel free to discuss strategies but not code with others.

- Your program must be based on **Python3**.
- You **must** use minimax search with alpha-beta pruning as the basic algorithm.
- You **must** use the skeleton code provided so that your program can use the user interface we provide and participate in the tournament.
- You must design your own heuristic functions and justify them in the report you turn in.
- You should experiment with different strategies for more efficient or better search from the book or of your own design (e.g. different heuristics, shallow search, searching deeper under a potential move).
- You should play against your AI, trying out your different variants and different max levels of search, and report on the results.
- Your report should include the following:
    - A brief description of the game.
    - The definition of your heuristic function and its justification. Illustrate how it works with example moves it chose and explain why it chose them in terms of the function.
    - The details of your program design, experiments, and results, which should include:
        - How long it takes to make a move at different search depths and specify your CPU speed in the report. Timing restrictions are only for the tournament.
        - How does each heuristic function you designed affect the performance, e.g., the algorithm with this heuristic function vs the algorithm without this heuristic function?
- This assignment is bigger than the previous two; writing code without debugging will lead to a disaster.
- You are welcome to play against each other using the Internet! The TAs have set up a server for Internet games (human vs. human or AI vs. AI)

Tips on implementation

System preparation:
To play Kalah, we suggest running the code locally before playing against a remote opponent.

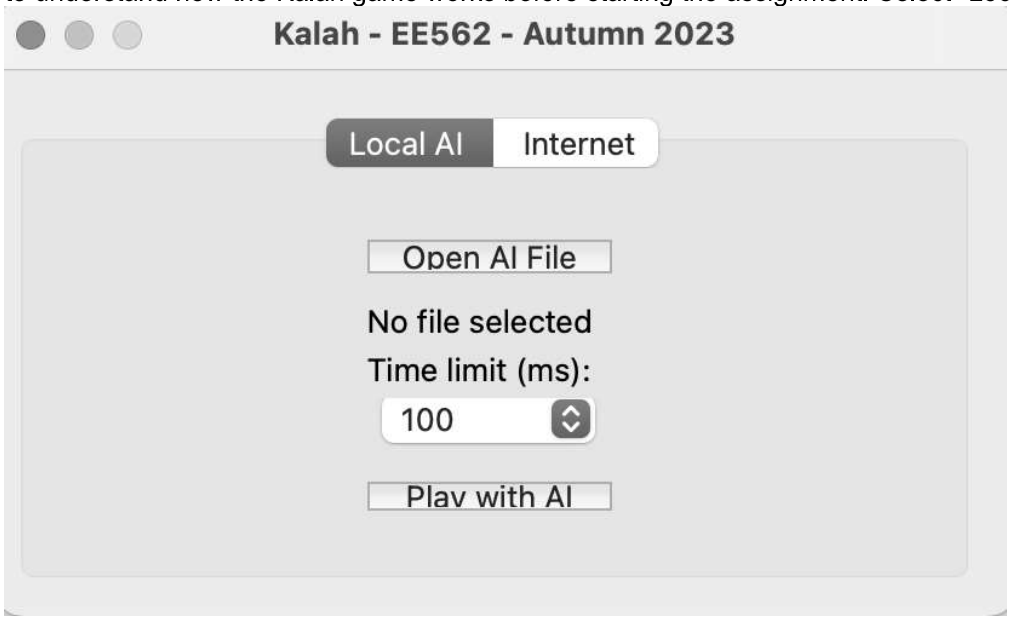
Conda (on Windows, Linux, Mac):
The first 3 steps are not strictly necessary but may make running the HW code easier.
SUGGESTED: Download the 3 hw files (main.py, ai.py, ui.py). Install suds-py3 and PySide6. Try running the code from within your homework directory. If the code runs -- don't install Anaconda (your current Python 3 installation is fine). If you want to work in a virtual environment, you can, but it is neither essential nor required.

1. Install Anaconda from here according to your platform.
2. Open cmd and create a virtual environment named "hw3" by running: "conda create -n hw3 python=3.x". Replace the x with your version of Python3.
3. Activate the virtual environment: "source activate hw3" ("conda activate hw3" on Windows).
4. Install suds: " pip install suds-py3". See here).
5. Install Qt: "pip install PySide6". See here).
6. Run **python main.py** in the skeleton package.

NOTE: The game server can only be connected when you are using Campus Internet.
If you are connecting from offsite, please install Husky OnNet (See here).

If you want to work on the code offsite, without connecting, comment out the lines referring to the URL of the server (lines #43 and #45 in main.py)

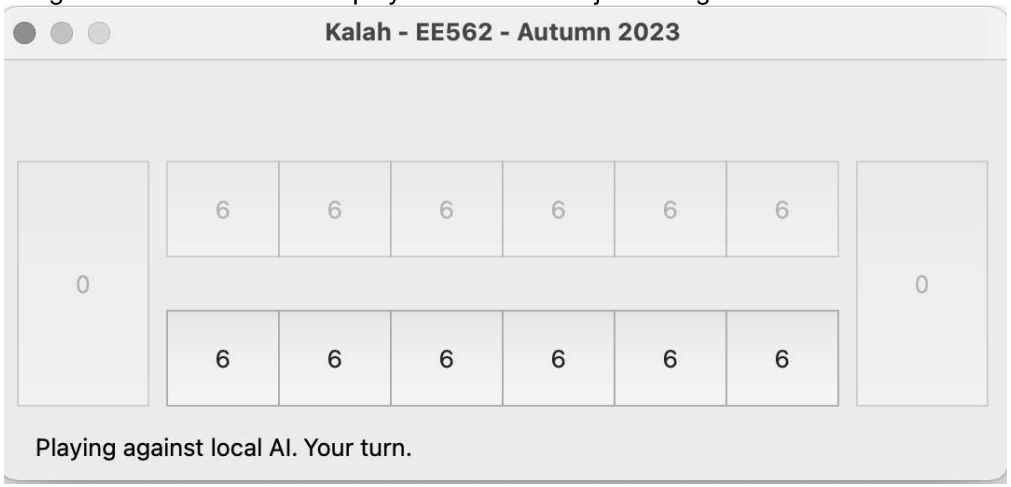
To use the UI:
1. Human vs. local AI.
You can create and play a game locally against the AI described in your ai.py file. The AI provided with the skeleton code is not very smart, but you can play it to understand how the Kalah game works before starting the assignment. Select "Local AI" followed by "Open AI File" and "Play with AI" to start a game.



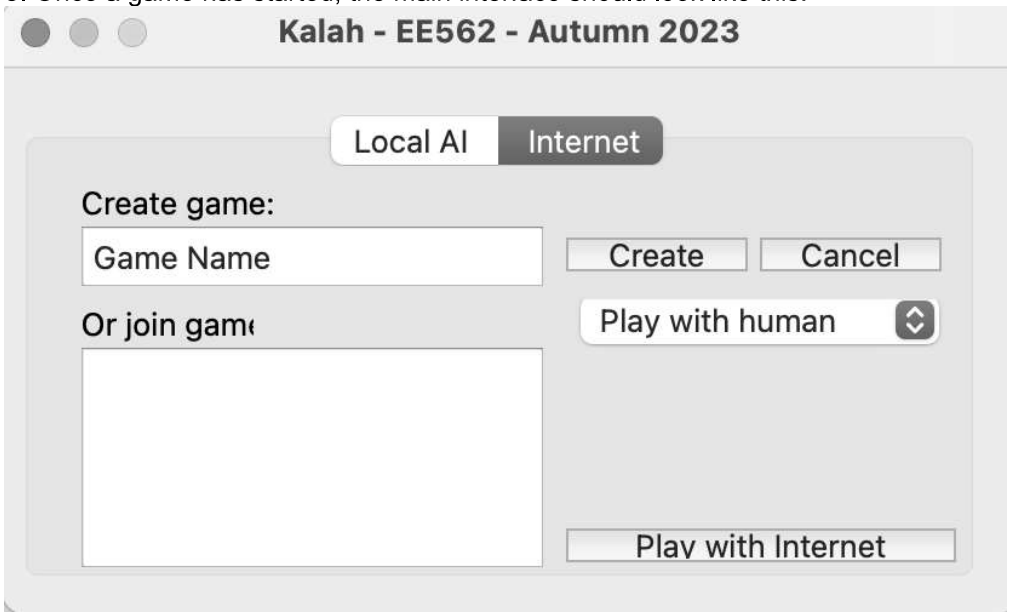
2. Human vs. internet human / Local AI vs. internet AI

NOTE: The code has been checked on Mac and Windows. If you are unable to make remote play work properly, the TAs will try to assist you. However, you can still complete the assignment even if you are only able to run the code locally on your machine.

Select "Play with human" or "Play with AI" when creating a host. To play via the Internet, the host should first create a game. Then a second player can select the game in the list and click 'play with internet' to join that game.



3. Once a game has started, the main interface should look like this:



A known bug of this framework is that occasionally the UI will become stuck and one or more buttons become grey, but the numbers are not 0. You just need to minimize the window and resume, which will be fine. Very seldom it will crash. Just open it again.

Implement your AI:
There's a Python script file ai.py in which you will implement your code. The other two should not be modified. There's a detailed explanation of input and output inside ai.py. Remember the requirements and time limitations mentioned above.

Note: The game server can only be connected when you are using Campus Internet. If you are connecting from offsite, please install Husky OnNet (See here).

Turn-in and Evaluation

You should turn in the following **two files** within a single .zip file:

- Your well-commented code (ai.py), 17 points:
  - Working program: 15 points.
  - Well-commented code: 2 points.
- Report describing your implementation (at most 5 pages, 1” margins, 11pt, double-spaced), 8 points:
  - A brief description of the game: 1 point.
  - The definition of your heuristic function and its justification. Illustrate how it works with example moves it chose and explain why it chose them in terms of the function: 3 points.
  - The details of your program design, experiments, and results:

- How long it takes to make a move at different search depths and specify your CPU speed in the report. Timing restrictions are only for the tournament: 2 points.
- How each heuristic function you designed affects the performance, e.g., the algorithm with this heuristic function vs the algorithm without this heuristic function: 2 points.

Tournaments

The TAs will run a Kalah tournament. This is mostly for fun, but the winners will be recognized and there will be some (small) extra points for the tournament winners (top 4).
Comparison: the same machine will be used for all the games in the Kalah tournament.
You can use whatever you know to improve your AI (multi-thread, special gaming strategy database, etc). The only limitation is 1 second in searching time on the TA's machine. The framework will time your AI execution; if you exceed 1 second, you will lose the game.