Distributed and Outsourced Software Engineering (DOSE 2015)

Assignment 3: Implementing a Game using the AI Search Library

(Mandatory)

Deadline First Iteration: Tuesday, November 17th 11pm (CET)
Final Deadline: Sunday, November 29th 11pm (CET)

1. Software Development Process

Development during the second phase of the DOSE project will be organized in two iterations. Similar to Assignment 2, the development has to follow the DOSE 2015 *Software Development Process.* The deadlines for this Assignment are:

- ***Deadline First Iteration: Tuesday, November 17th 11pm (CET)***
- ***Deadline Second Iteration: Sunday, November 29th 11pm (CET)***

2. GitHub Projects and Forks

2.1 Forks

Each group has already forked the dose2015 project. The development of this second phase will be performed within each group's fork, using the search engines already developed by the group during the first phase. As for the first phase, you are not allowed to give access to your fork to students outside your group.

2.2 Source code folder for this assignment

Each group has a folder:

/03_games/group***i*** where ***i*** is the group number

Within the group's fork, each group member must only modify the folder of the group he/she belongs to. This is important, so we can merge the forks later on without conflicts.

2.3 Project Iterations and Planning

This assignment is organized in two iterations. The corresponding tasks for each iteration must be defined by the group members, in coordination and with strong participation from the SW PL. The SW PL, together with the group members, has to plan each iteration with a list of tasks. We strongly suggest the SW assigns tasks to group members and each group member then devises a number of activities required for achieving his/her assigned tasks. These activities must be formalized as issues in the project's fork issue tracker (including the team members' assignments).

Task: Before each iteration, add the tasks (for the upcoming iterations) to the GitHub issue tracker. Assign who is responsible for each task. Tag the tasks of different iterations with the following labels:

app-iteration1, ***app-iteration2***: tasks in Assignment 3

2.4 Wiki

After each iteration, the SW PL has to create a Wiki page (called Iteration Report):

- Number of open and closed issues
- Number of new additional tasks / issues created in the iteration
- Number of tasks / issues closed during the iteration
- For each group member:
  - Number of tasks closed by the group member
  - Number of open issues assigned to the group member

Task: After each iteration, add a link to the Iteration Report in the DOSE Wiki page:

https://github.com/DOSE-ETH/dose2015/wiki/Groups

3. Project

The project to implement during this assignment is a Mancala game. The game application shall feature:

- A graphical user interface for the user to interact with the game.
- The application shall allow the user to play in two modes:
  - ***Solitaire mode:*** it shall follow the game rules as described later on in this document (adapted from [1]). This mode shall provide two additional AI features: *game hint* (suggesting a move to the player) and *automatic play* (the computer plays in solitaire mode, attempting to win/solve the game).
  - ***Adversary mode:*** it shall follow the game rules described in the tutorial in [2]. This mode shall allow a user to play Mancala against the computer.
- ***Solitaire mode*** and ***adversary mode*** shall allow the user to select the engine to be used for AI (***solitaire mode*** uses single-agent search, whereas ***adversary***

mode uses adversary search engines), with the possibility of setting -for some engines- its corresponding parameters (e.g., max depth for bounded engines). These settings may be provided application-wide, i.e., a separate menu item allowing the user to set engines and engine parameters, prior to starting a game.

Your group will have to self-organize how you split the workload between the all team members.

We strongly suggest all group members to familiarize themselves with the game, since the development will require implementing heuristic functions and other search related mechanisms that require a good understanding of the game. An online version of the game in adversary mode is available at [2]. Notice that there exist many rule variants for Mancala. Make sure your implementation follows those described in 3.1 and those of the tutorial in [2].

3.1 Solitaire Mancala Rules

SR001: Initial Setup – Mancala board. The game shall be played using a standard Mancala board consisting of two rows, where each row has 6 holes and 1 store. The board is defined as:

store01	hole01	hole02	hole03	hole04	hole05	hole06	store02
	hole07	hole08	hole09	hole10	hole11	hole12	

SR002: Initial Setup – initial configuration. The game shall be initiated with 48 stones randomly split into the 12 holes (hole01, hole02, … hole12; see SR001), leaving each hole with at least 1 stone (all 48 stones shall all be placed in the board, leaving the stores empty).

SR003: Selection of hole before first round*.* The player shall select a hole (in any row, thus any of the 12 holes).

SR004: A round. When a hole is selected, all stones in the selected hole are distributed either clockwise or anti-clockwise (the user can choose), as follows:

> ***SR004(A): Dropping clockwise.*** One stone is placed in each hole starting with the hole next to the selected one, in clockwise direction:
>
> - If the number of stones remaining to be distributed is more than 1 after dropping in hole06 or hole07, then store02 or store01 respectively is skipped, and the next stone is dropped in hole12 or hole01 respectively. The round is over when there are no more stones to distribute. If the *game is not over* (see SR005 and SR006), a new round starts as

described in SR004, where the selected hole will be the one where the last stone of this round was dropped.

- If the number of stones remaining to be distributed is 1 after dropping in hole06, this stone is dropped in store02. The round is over. If the *game is not over* (see SR005 and SR006), a new round starts as described in SR004, where the selected hole will be hole06.
- If the number of stones remaining to be distributed is 1 after dropping in hole07, this stone is dropped in store01. The round is over. If the *game is not over* (see SR005 and SR006), a new round starts as described in SR004, where the selected hole will be hole07.

SR004(B): Dropping anti-clockwise. One stone is placed in each hole starting with the hole next to the selected one, in anti-clockwise direction:

- If the number of stones remaining to be distributed is more than 1 after dropping in hole01 or hole12, the store01 or store02 respectively is skipped, and the next stone is dropped in hole07 or hole06 respectively. The round is over when there are no more stones to distribute. If the *game is not over* (see SR005 and SR006), a new round starts as described in SR004, where the selected hole will be the one where the last stone of this round was dropped.
- If the number of stones remaining to be distributed is 1 after dropping in hole01, this stone is dropped in store01. The round is over. If the *game is not over* (see SR005 and SR006), a new round starts as described in SR004, where the selected hole will be hole01.
- If the number of stones remaining to be distributed is 1 after dropping in hole12, this stone is dropped in store02. The round is over. If the *game is not over* (see SR005 and SR006), a new round starts as described in SR004, where the selected hole will be hole12.

SR005: Game over – player loses. When the last stone distributed in a round is placed in an empty hole, the player loses and the game is over.

SR006: Game over – player wins. The player wins the game if no stone remains in any of the 12 holes.

4. Final Task: pull request

Once the group has finished the assignment, the SW Project Leader has to send a *Github Pull Request* [3] to the original DOSE repository (https://github.com/DOSE-ETH/dose2015/). We will merge the project on Monday November 30th; for this reason, we kindly request at least one member of the group to be available on Monday November 30th in case we have problems to merge. Before sending the pull request, please check carefully that you only modified files located in your group folders.

5. References

[1] Rules of Solitaire Mancala. http://mancala.wikia.com/wiki/Mancala_Solitaire_(1)

[2] Free Online Mancala Game. http://play-mancala.com

[3] Github Pull Request https://help.github.com/articles/using-pull-requests